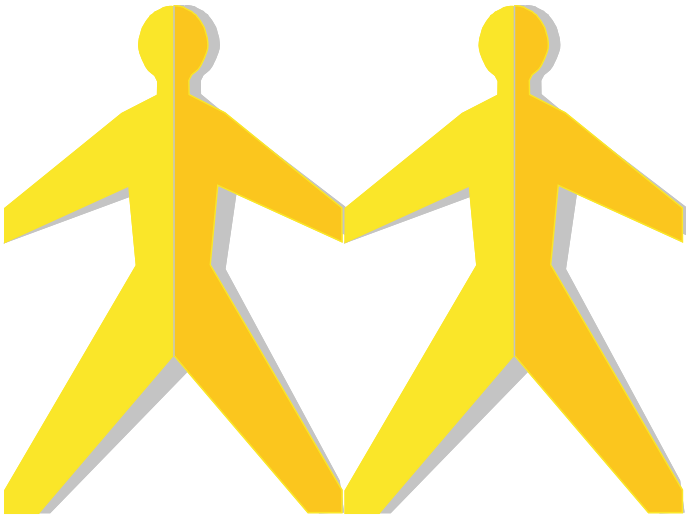
gedas



rvs® Tiny

Version 2.0

Benutzerhandbuch

Die in diesem Handbuch aufgeführten Produkte sind urheberrechtlich geschützt und stehen dem jeweiligen Rechtsinhaber zu.

rvs® Tiny
Version 2.0
Benutzerhandbuch

© 2004 by gedas deutschland GmbH
Pascalstraße 11
10587 Berlin

Das vorliegende Handbuch ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Kein Teil dieses Buches darf ohne Genehmigung von gedas in irgendeiner Form durch Fotokopie, Mikrofilm oder andere Verfahren reproduziert oder in eine für Maschinen, insbesondere Datenverarbeitungsanlagen, verwendbare Sprache übertragen werden. Auch die Rechte der Wiedergabe durch Vortrag, Funk und Fernsehen sind vorbehalten.

Inhaltliche Änderungen dieses Handbuches behalten wir uns ohne Ankündigung vor. gedas haftet nicht für technische oder drucktechnische Fehler oder Mängel in diesem Handbuch. Außerdem übernimmt gedas keine Haftung für Schäden, die direkt oder indirekt auf Lieferung, Leistung und Nutzung dieses Materials zurückzuführen sind.

Inhalt

1 Einführung 3

1.1 Kurze Beschreibung des Systems 3
1.2 Repräsentationsmittel 6
1.3 Zielgruppe 6

2 Installation auf Windows-Systemen 7

2.1 Systemvoraussetzungen 7
2.2 Erhalten einer Lizenz 7
2.3 Neuinstallation von rvs® Tiny 8
2.4 rvs® Tiny als Windows-Dienst 13
2.5 Wie starte ich die rvs® Tiny - Benutzeroberfläche? 14
2.6 Update-Installation von rvs® Tiny 15

3 Konfiguration 17

3.1 Anpassen der globalen Konfigurationsdateien 17
3.1.1 rvsTiny.properties 17
3.1.2 rvsTinyConfig.xml 18
3.2 Anpassen der Stationskonfiguration 19
3.2.1 Konfiguration von Stationen mittels GUI 20
3.2.2 Konfiguration von Stationen mittels der XML-Datei 23
3.2.3 Identifikation von rvs® Tiny Stationen 24
3.2.4 Stationseintrag STATION_LOC 27
3.2.5 Stationseintrag STATION_NK 28
3.2.6 Stationseintrag STATION_RT 29
3.3 Konfiguration der JobStarts 30

4 Arbeiten mit rvs® Tiny 34

4.1 Starten des rvs® Tiny Servers 34
4.2 Stoppen des rvs® Tiny Servers 34
4.3 Meldungen des Monitors anzeigen 35
4.4 Aktivieren einer Station 37
4.5 Versand einer Datei 38
4.6 Synchronisation von Sendeaufträgen 45
4.7 Ausgabe aller Empfangs- und Sendejobs 49
4.8 Ausgabe eines JobEintrages 53
4.9 Löschen oder Freigeben von EERPs 54
4.10 Archivieren der Einträge der abgearbeiteten Sende- bzw. Empfangsaufträge im Revisionslog 54

5 Zentrale Administration von rvs® Tiny 55

5.1 Einleitung 55
5.2 Programme der Zentralen Administration 57
5.3 Wie arbeite ich mit der Zentralen Administration? 58
5.3.1 Wie tausche ich einen Lizenzschlüssel aus? 60
5.3.2 Wie ändere ich die Konfiguration einer Station? 61
5.3.3 Wie führe ich ein Update von rvs® Tiny durch? 63
5.4 Wie verteile ich rvs® Tiny Installationen? 64

© gedas deutschland GmbH / Fachbereich rvs® Systems

1 Einführung

Dieses Kapitel beinhaltet eine kurze Beschreibung der Produkte rvs® und rvs® Tiny, sowie eine Erklärung der typografischen Auszeichnungen, die in diesem Handbuch verwendet werden.

1.1 Kurze Beschreibung des Systems

Was ist rvs®

rvs® = Rechner-Verbund-System

Die Abkürzung rvs® steht für die Bezeichnung Rechner-Verbund-System. Das rvs® Rechner-Kommunikations-System ist ein etablierter Basisdienst für elektronischen Datenaustausch, EDI.

rvs® hat die Aufgabe, die Übertragung von elektronischen Daten zwischen heterogenen Computersystemen zu gewährleisten, die unterschiedliche Netzwerkprotokolle verwenden.

Um das zu erreichen, verwirklicht rvs® ein universelles Netzwerkmodell, das Sie für jeden Netzwerkknoten konfigurieren können.

rvs® sorgt für einen zuverlässigen und leistungsfähigen Transportdienst für standardisierte EDI Nachrichten und für Dateien mit beliebigem Format und Inhalt. Sie können nur Dateien empfangen, die für rvs® vorgesehen sind. Das bedeutet, dass rvs® keinen unautorisierten Zugang zu anderen oder den eigenen Daten zulässt.

Das System wurde ursprünglich von der Volkswagen AG entwickelt und wird seit mehreren Jahren in der deutschen und europäischen Automobilindustrie genutzt, ist aber auch weltweit bei Banken, Versicherungen und in der Industrie im Einsatz.

rvs® arbeitet mit dem OFTP Protokoll. Für die Volkswagen AG wurde eine Erweiterung zum OFTP Standard entwickelt: Es wurde um einen Leitungstreiber für SNA LU 6.2 erweitert.

Was rvs® nicht ist

rvs® ist kein Onlinesystem. Es unterstützt weder den direkten terminalähnlichen Zugang zu anderen Rechnern, noch ermöglicht es eine Kommunikations-Pipe von Anwendung zu Anwendung auf Datensatz-Ebene. Sie können keine direkte Übertragung Ihrer Daten aus der Anwendung heraus ausführen. Sie können jedoch Sendeaufträge aus der Anwendung heraus an rvs® übergeben, die dann asynchron ausgeführt werden.

rvs® ist kein System, das Arbeitsaufgaben einplant.

rvs® interessiert sich nicht für den Inhalt der Dateien, die es überträgt. Es funktioniert als nachvollziehbares Transportmedium und führt keine Bedeutungsinterpretation der Daten durch, die es übermittelt.

rvs® ist kein EDI Konverter. Jedoch sind zusätzliche Komponenten zur Konvertierung zwischen spezifischen Nachrichtenformaten (z.B. VDA,

© gedas deutschland GmbH / Fachbereich rvs® Systems

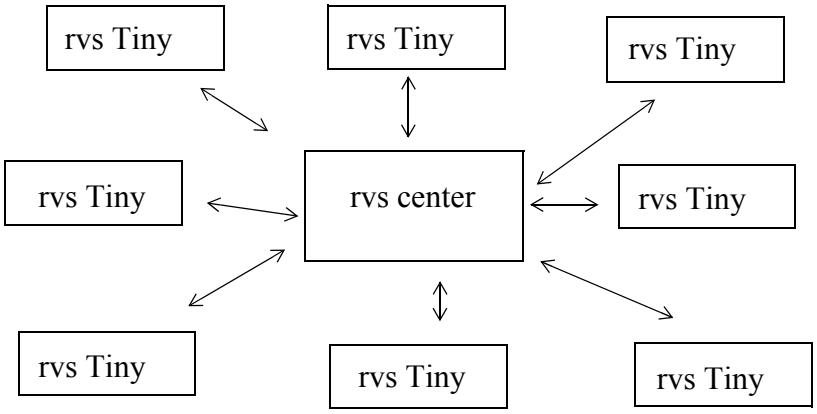
ODETTE, EDIFACT, XML), die rvs® als Transportdienst benutzen, bei der gedas deutschland GmbH verfügbar.

rvs® ist keine Software zur Netzwerksteuerung oder Überwachung.

Was ist rvs® Tiny

rvs® Tiny

rvs® Tiny ist eine Kommunikationssoftware mit der grafischen Benutzeroberfläche, die wie rvs®, auf dem OFTP-Protokoll basiert. Sie ist in der Lage, Applikationen mit Partnern zu verbinden. Daher kommt diese Software in Sterntopologien zum Einsatz, in denen eine Vielzahl von Filialen oder mobilen Mitarbeitern mit einem zentralen Server verbunden ist.



Folgende Funktionalitäten stehen bei rvs® Tiny 2.0 zur Verfügung:

Funktionalitäten

- grafische Benutzeroberfläche (GUI).
- Versand von Dateien an die rvs® - Zentrale oder an geroutete Stationen (die über rvs® - Zentrale erreichbar sind).
- Empfang von Dateien von der rvs® - Zentrale oder von gerouteten Stationen (die über die rvs®-Zentrale erreichbar sind).
- Aktivierung der rvs® - Zentrale um abholbereite Dateien zu empfangen oder die Verbindung zu testen.
- Zentrale Administration von anderen rvs® Tiny Installationen.
- Anzeige von Informationen über aktive Empfangs- und Sendeübertragungen, über beendete und über fehlerhafte Übertragungen.
- Verfolgen von Monitoraktivitäten und Unterstützung bei der Fehleranalyse mit Hilfe der rvs® Tiny-Logdateien.
- Löschen oder Freigeben von Empfangsbestätigungen (EERPs).

– Codeumwandlung (ASCII -> EBCDIC und EBCDIC -> ASCII) mit verschiedenen Umwandlungstabellen.
– Formatumwandlung während der Übertragung (unterstütze Formate: Fixed, Variabel, Text, Unstrukturiert).
– Jobstarts, um die gewünschten Aktionen beim Empfang oder Versand anzustoßen.

rvs® Tiny kommuniziert mit der Applikation über eine Batch-Schnittstelle sowie über das Dateisystem. Die angebundene Applikation kann, sofern sie in der Lage ist, die erfolgreiche Weiterverarbeitung signalisieren und den erfolgreichen Versand signalisiert bekommen.

rvs® Tiny ist implementiert in Java.

TCP/IP Protokoll

Z.Zt. unterstützt rvs® Tiny nur das OFTP-TCP/IP Protokoll.

Unterstützte Plattformen

rvs® Tiny ist eine Java-Applikation und daher plattformunabhängig.

Zur Zeit stehen Installationsroutinen für die folgenden Plattformen zur Verfügung:

– rvs® Tiny für MS Windows XP mit TCP/IP
– rvs® Tiny für MS Windows 2000 mit TCP/IP

Wenn Sie eine Installationsroutine für eine andere Plattform wünschen, wenden Sie sich bitte an:

rvs® - Kundendienst

Telefon: +49-30-39 97-17 77

Telefax: +49-30-39 97-19-94

e-mail: rvs-service@gedas.de

© gedas deutschland GmbH / Fachbereich rvs® Systems

1.2 Repräsentationsmittel

Dieser Abschnitt enthält die Beschreibung, welche Ausprägungen und Auszeichnungen in diesem Handbuch verwendet werden und welche Bedeutung besonders gekennzeichnete Ausdrücke haben.

Typografische Auszeichnungen

- Handlungsanweisungen beginnen mit dem Punkt als Aufzählungszeichen.
- – Sonstige Aufzählungen verwenden den Halbgeviertstrich.

Zeichenformate

`Courier`	Kommandos, Menübefehle, Dateinamen, Pfadnamen, Programme, Beispiele, Script-Dateien, Optionen, Qualifiers, Datensätze, Felder, Modi, Fensternamen, Dialogboxen und Status
FETT und **GROSSBUCHSTABIG**	Parameter, Umgebungsvariablen, Variablen
"Hochkommata"	Verweise auf andere Handbücher, Kapitel und Abschnitte, Literatur
Fett	wichtige Begriffe, Betriebssystemnamen, Eigennamen, Schaltflächen (Buttons), Funktionstasten.

Verzeichnisse

$RVSTINY_HOME

Weil Benutzerverzeichnisse auf unterschiedlichen Plätzen bei den unterschiedlichen Betriebssystemen zu finden sind, benutzen wir in diesem Handbuch die Variable **$RVSTINY_HOME**. Die Standardwerte sind:

– `C:\Programme\rvsTiny` für **Windows XP** und **Windows 2000**.

Ersetzen Sie diese Variable durch Ihren richtigen Pfad.

1.3 Zielgruppe

Dieses Handbuch ist sowohl für Benutzer gedacht, die mit rvs® Tiny routinemäßig arbeiten, als auch für Administratoren. Es gibt Ihnen einen Überblick über die Basisfunktionalität von rvs® Tiny.

Kenntnisse

Folgende Fähigkeiten sind erforderlich, um rvs® Tiny nutzen zu können:

– gute Kenntnisse über das benutzte Betriebssystem

– Kenntnisse über aktuell verwendete Kommunikationstechniken TCP/IP

Wir empfehlen dieses Handbuch zu lesen, bevor Sie anfangen mir rvs® Tiny zu arbeiten.

2 Installation auf Windows-Systemen

In diesem Kapitel werden die Installationsvoraussetzungen für Windows-Systeme und der Installationsvorgang von rvs® Tiny beschrieben.

2.1 Systemvoraussetzungen

Für einen erfolgreichen Betrieb von rvs® Tiny benötigen Sie folgende Software:

Software

– Betriebssystem: Windows XP oder Windows 2000.

– Java Laufzeitumgebung (JRE 1.3.1_XX oder Java Software Development Kit 1.3.1_XX).

  Falls auf ihrem System noch keine Java Laufzeitumgebung installiert ist, dann holen Sie dies bitte vor der Installation von rvs® Tiny nach. Die Software ist frei verfügbar. Sie können diese über die Java Webseite http://java.sun.com herunterladen.

Am Anfang brauchen Sie mindestens 8 MB freien Platz auf Ihrer Festplatte. Abhängig von der Gebrauchsintensität, der Erhaltungsdauer für alte Einträge und der Zeitspanne zwischen Datenbankbereinigungen können die Speicherplatzanforderungen erheblich größer sein.

Zur Zeit unterstützt rvs® Tiny nur das OFTP-TCP/IP Protokoll.

In der Regel wird rvs® Tiny als CD ROM oder Datenband geliefert. Ihr System muss diese Datenträger lesen können. Wenden Sie sich bitte an Ihren Vertriebspartner, wenn Sie andere Voraussetzungen haben.

2.2 Erhalten einer Lizenz

Um mit rvs® Tiny arbeiten zu können, benötigen Sie einen Lizenzschlüssel.

rvs® - Kundendienst

Um einen Lizenzschlüssel zu erhalten, wenden Sie sich bitte an den rvs® - Kundendienst (Telefon: +49 30 39971 777; Telefax: +49 30 39971 994; E-Mail: rvs-service@gedas.de).

Folgende Schritte sind notwendig, um einen Lizenzschlüssel zu erwerben:

– In dem Eingabeaufforderungsfenster (`Ausführen -> cmd`) den Befehl `hostname` eingeben.

– Die Ausgabe dieses Befehls (Ihr Rechnername z.B. `BWcd00034`) ist dem rvs® - Kundendienst mitzuteilen. Beachten Sie dabei unbedingt die Groß-/Kleinschreibung, da andernfalls kein gültiger Lizenzschlüssel für Ihren Rechner generiert werden kann.

– Ihre Lizenzschlüsseldatei erhalten Sie per E-Mail.

Die Lizenzschlüsseldatei muss im Ordner `$RVSTINY_HOME\conf\` unter dem Dateinamen `license.properties` gespeichert werden.

© gedas deutschland GmbH / Fachbereich rvs® Systems

Hinweis: Lesen Sie bitte das Kapitel 1.2 "Repräsentationsmittel" für eine Erklärung von `$RVSTINY_HOME`.

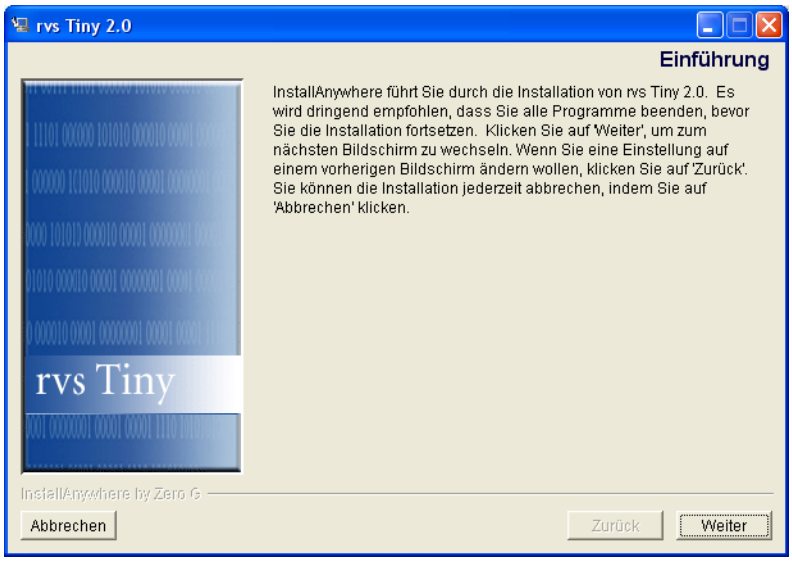
2.3 Neuinstallation von rvs® Tiny

Installationsschritte

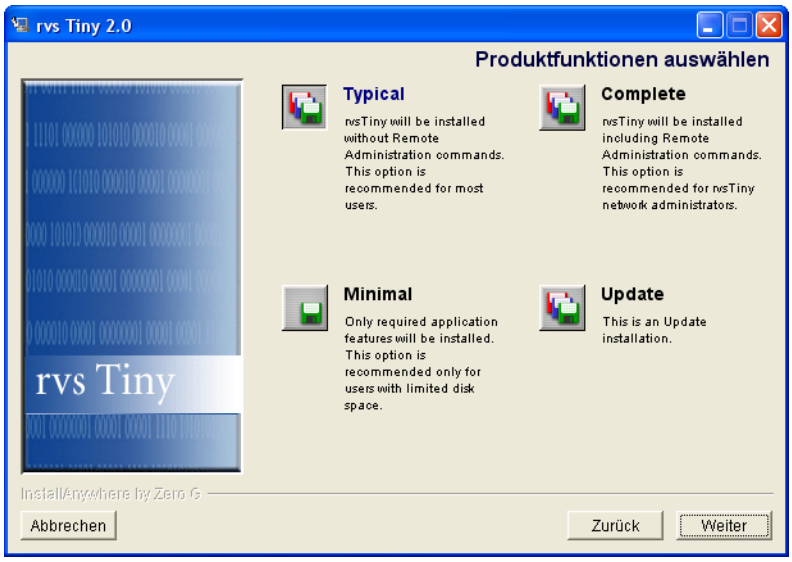
In diesem Kapitel wird die Installation von rvs® Tiny beschrieben. Bevor Sie mit der Installation beginnen, vergewissern Sie sich, dass Ihre Systemumgebung alle Voraussetzungen für eine erfolgreiche Installation erfüllt (Siehe Kapitel 2.1 "Systemvoraussetzungen").

- Starten Sie Ihr Windows-System und melden Sie sich als Windows-Benutzer mit Administrator-Rechten an.
- Starten Sie die Installationssoftware `rvsTiny_X.X_setup.exe` (wobei `X.X` der Versionsnummer von rvs® Tiny entspricht) per Doppelklick oder über den Windows-Befehl: `Start -> Ausführen`.
- Im ersten Dialog können Sie die Sprachversion der Installation auswählen (Deutsch, Englisch, Spanisch…). Mit **<OK>** kommen Sie zum nächsten Schritt der Installationsroutine.

- Die zweite Bildschirmanzeige gibt Ihnen eine kleine Einführung in den Installationsprozess.

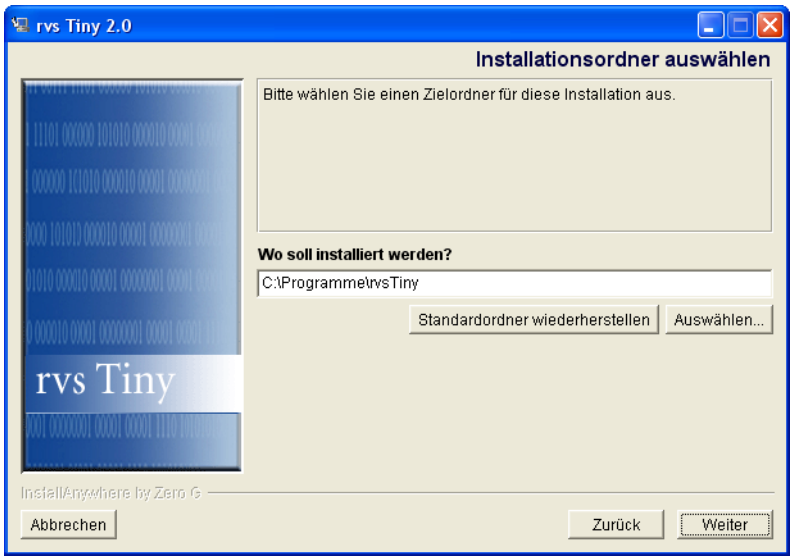

rvs Tiny 2.0
Einführung
InstallAnywhere führt Sie durch die Installation von rvs Tiny 2.0. Es wird dringend empfohlen, dass Sie alle Programme beenden, bevor Sie die Installation fortsetzen. Klicken Sie auf 'Weiter', um zum nächsten Bildschirm zu wechseln. Wenn Sie eine Einstellung auf einem vorherigen Bildschirm ändern wollen, klicken Sie auf 'Zurück'. Sie können die Installation jederzeit abbrechen, indem Sie auf 'Abbrechen' klicken.
rvs Tiny
InstallAnywhere by Zero G
Abbrechen
Zurück
Weiter

- Im nächsten Dialog können Sie eine „Typical“, „Complete“, „Minimal“-Installation oder ein Update wählen. Eine „Complete“Installation brauchen Sie, wenn Ihre rvs® Tiny-Installation andere rvs® Tiny Installationen administrieren soll (siehe Kapitel 5 "Zentrale Administration von rvs® Tiny").

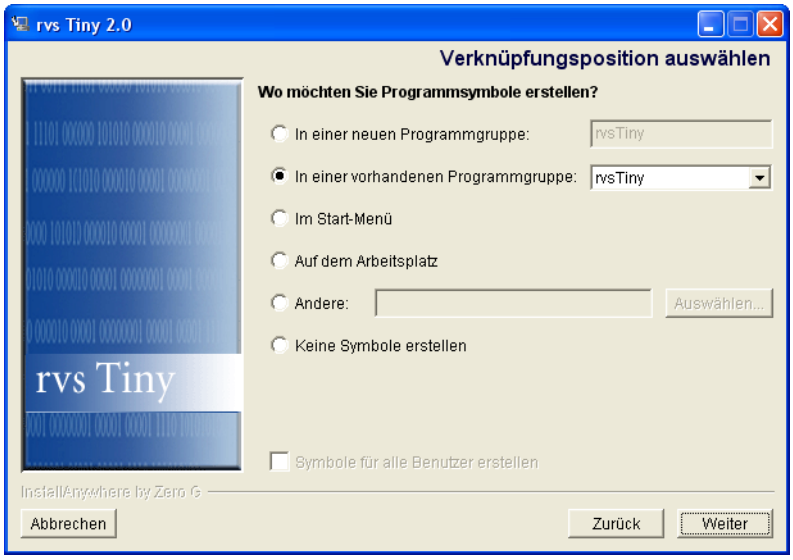

rvs Tiny 2.0
Produktfunktionen auswählen
Typical
rvsTiny will be installed without Remote Administration commands. This option is recommended for most users.
Complete
rvsTiny will be installed including Remote Administration commands. This option is recommended for rvsTiny network administrators.
Minimal
Only required application features will be installed. This option is recommended only for users with limited disk space.
Update
This is an Update installation.
rvs Tiny
InstallAnywhere by Zero G
Abbrechen
Zurück
Weiter

- Im vierten Dialog können Sie das Verzeichnis auswählen, in das rvs® Tiny installiert werden soll.

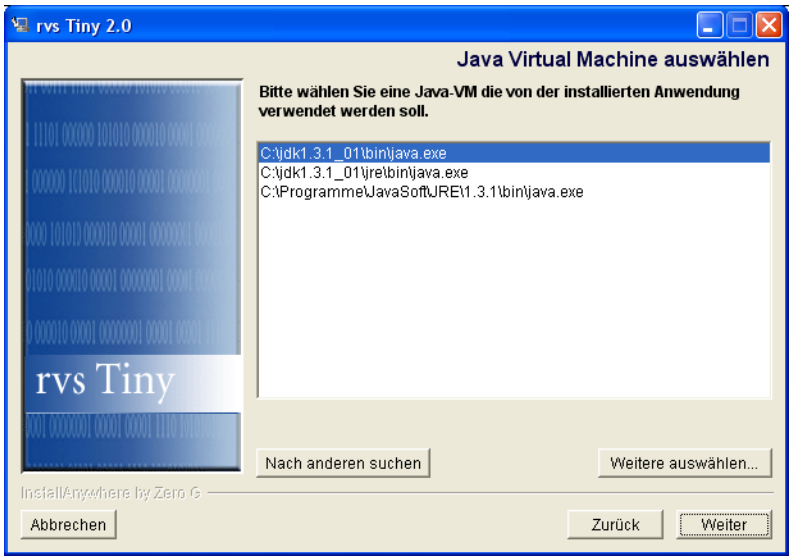
© gedas deutschland GmbH / Fachbereich rvs® Systems



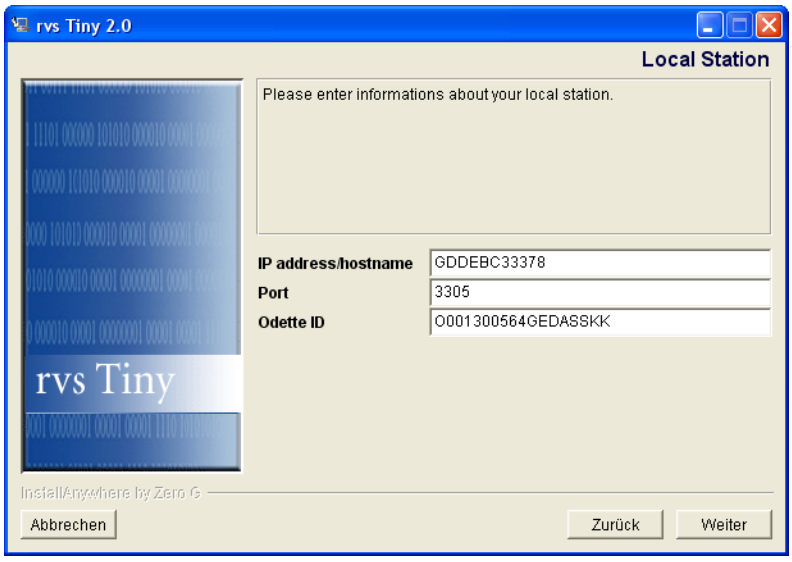
- Im fünften Dialog legen Sie fest, in welcher Programmgruppe die Symbole für rvs® Tiny erstellt werden sollen.



- Im nächsten Dialog legen Sie fest, welche Java - Laufzeitumgebung für den Betrieb von rvs® Tiny eingesetzt werden soll. Hierbei sucht das Installationsprogramm nach installierten Komponenten und schlägt Ihnen die gefundenen Versionen in einem Auswahldialog vor. rvs® Tiny ist für einen Betrieb für die Version 1.3.1_X freigegeben.

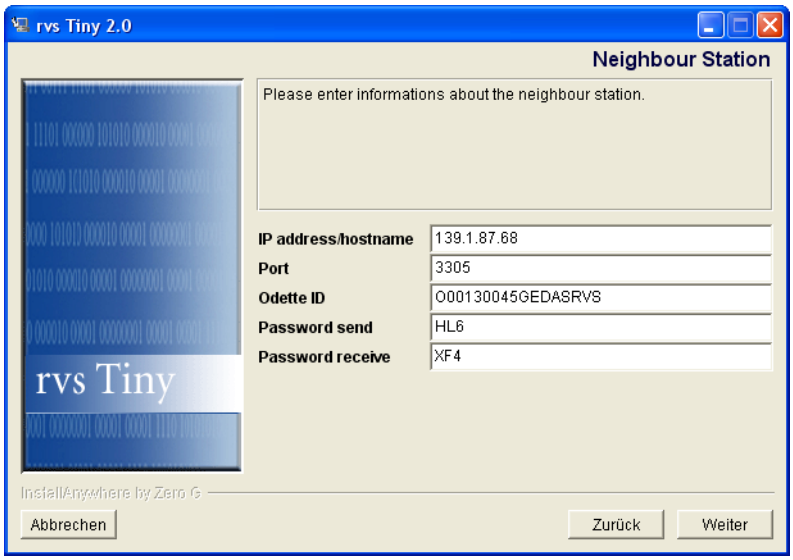


- Die nächsten Dialoge kümmern sich um die Einstellungen in der Stationsliste. Zunächst müssen Sie den Eintrag für die lokale Station konfigurieren. Die drei Parameter im Dialog sind obligatorisch und müssen eingegeben werden. Lesen Sie bitte das Kapitel 3.2.3 "Identifikation von rvs® Tiny Stationen" für ausführliche Informationen über die Stationsparameter.



- Danach folgt der Dialog für die Eintragungen der direkten Nachbarstation (rvs®-Zentrale). Auch hier sind alle Parameter obligatorisch und müssen eingegeben werden, um reibungsloses Arbeiten von rvs® Tiny zu gewährleisten. Lesen Sie bitte das Kapitel 3.2.3 "Identifikation von rvs® Tiny Stationen" für ausführliche Informationen über die Stationsparameter.

© gedas deutschland GmbH / Fachbereich rvs® Systems



- Die letzte Bildschirmanzeige informiert Sie über die erfolgreiche Installation von rvs® Tiny.

Start und Stopp

- Um zu testen, ob die Installation erfolgreich war, starten und stoppen Sie rvs® Tiny mit folgenden Befehlen:

  `startServer` / zum Starten des rvs® Tiny Prozesses.
  `stopServer` / zum Terminieren des rvs® Tiny Prozesses.

Diese beiden Befehle finden Sie jeweils in Form einer Batch-Datei im Verzeichnis `$RVSTINY_HOME\bin`.

Start und Stopp über die Eingabeaufforderung:

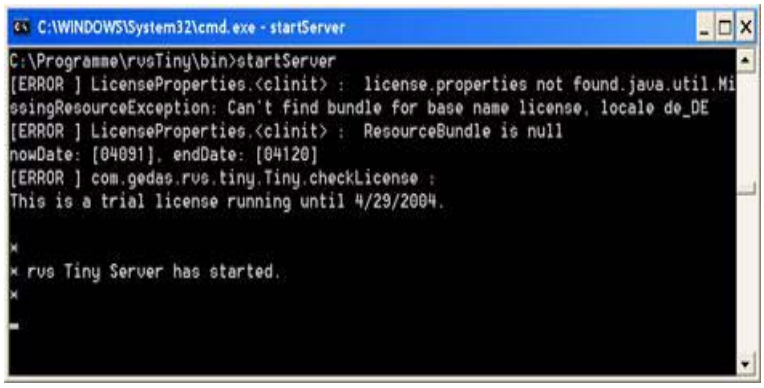
Alternativ können Sie rvs® Tiny aus der Eingabeaufforderung heraus starten oder stoppen. In diesem Falle müssen Sie sich im Verzeichnis `$RVSTINY_HOME\bin` (z.B. `C:\Programme\rvsTiny\bin`) befinden.

Erfolgreicher Start:

Start

Ein erfolgreicher Start sieht dann folgendermaßen aus:

:


C:\WINDOWS\System32\cmd.exe - startServer
C:\Programme\rvsTiny\bin>startServer
[ERROR] LicenseProperties.<clinit> : license.properties not found.java.util.Mi
ssingResourceException: Can't find bundle for base name license, locale de_DE
[ERROR] LicenseProperties.<clinit> : ResourceBundle is null
nowDate: [04091], endDate: [04120]
[ERROR] com.gedas.rvs.tiny.Tiny.checkLicense :
This is a trial license running until 4/29/2004.
rvs Tiny Server has started.

Start und Stopp über das Startmenü:

Starten Sie rvs® Tiny indem Sie im Startmenü `Start -> Programme -> rvsTiny -> rvsTiny` (oder den anderen, von Ihnen während der Installation festgelegten Startmenüeintrag) auswählen. Auf diesem Weg gelangen Sie ins Verzeichnis `$RVSTINY_HOME\bin`, wo Sie mit Doppelklick die beiden Batchdateien (`startServer.bat`; `stoperver.bat`) starten können.

2.4 rvs® Tiny als Windows-Dienst

Manchmal ist es wünschenswert, dass rvs® Tiny als Hintergrundprozess läuft, auch wenn der Benutzer nicht eingeloggt ist. Für diesen Zweck können Sie rvs® Tiny als Windows-Dienst installieren.

Hinweis: Der Begriff Dienst bedeutet ein Programm, das vom Betriebssystem aus gestartet werden kann und im Hintergrund arbeitet.

Die Installation von rvs® Tiny als Windows-Dienst ermöglicht Ihnen das Batch-Programm `rvstinyservice.bat`, welches sich im Verzeichnis `$RVSTINY_HOME\bin` befindet.

Syntax:

```
rvstinyservice <parameters> [options]
```

Mögliche Parameter sind:

`-c`	startet rvs® Tiny in der Eingabeaufforderung
`-i`	installiert rvs® Tiny als Windows-Dienst
`-r`	deinstalliert rvs® Tiny als Windows-Dienst
`-s`	startet den Windows-Dienst rvs® Tiny

© gedas deutschland GmbH / Fachbereich rvs® Systems

-h Hilfe

Beispiel:

Mit dem Befehl `rvstinyservice -c` starten Sie den rvs® Tiny Server in der Kommanozeile (Eingabeaufforderung).

Mit dem Befehl `rvstinyservice -i` installieren Sie rvs® Tiny als Windows-Dienst.

Nach der Installation von rvs® Tiny als Dienst können Sie rvs® Tiny in der Liste der Windows-Dienste finden (`Start -> Systemsteuerung -> Verwaltung -> Dienste`). Wenn Sie möchten, dass rvs® Tiny als Dienst bei jedem Systemstart automatisch gestartet wird, können Sie die Startart auf `Automatisch` setzen, indem Sie auf die Schaltfläche **Startart...** klicken und die Startart `Automatisch` wählen.

Beispiel:

2.5 Wie starte ich die rvs® Tiny - Benutzeroberfläche?

Starten Sie die rvs® Tiny - Benutzeroberfläche (GUI) mit dem Programm `startGUI`. Dieses Programm befindet sich im Verzeichnis `$RVSTINY_HOME\bin`, so dass Sie sich in diesem Verzeichnis befinden müssen, um das Programm ausführen zu können.

Die rvs® Tiny - Benutzeroberfläche können Sie auch mit Hilfe der rvs® Tiny - Programmgruppe starten (`Start -> Alle Programme -> rvs Tiny -> rvsTiny GUI`).

Syntax:

```
startGUI [-local]
```

Optionale Parameter:

`-local`	startet den OFTP-Server-Modul im gleichen Prozess wie die GUI; keine Übertragung von Daten ist möglich ohne, dass die GUI läuft; ein schon aktiver Serverprozess kann zu Fehlern führen.
`-help`	Hilfe
`-?`	Hilfe

Beispiel:

```
startGUI
```

Lesen Sie bitte das Kapitel 3 "Konfiguration", um mehr über die von der rvs® Tiny - Benutzeroberfläche unterstützen Funktionen zu erfahren.

2.6 Update-Installation von rvs® Tiny

Der Ablauf einer Update - Installation von rvs® Tiny ist fast identisch mit dem Ablauf einer normalen Installation.

Lesen Sie bitte das Kapitel 5 "Zentrale Administration von rvs® Tiny", um zu erfahren, wie Sie mit Hilfe der Zentralen Administration andere rvs® Tiny - Installationen updaten können.

© gedas deutschland GmbH / Fachbereich rvs® Systems

3 Konfiguration

Dieses Kapitel beschreibt, wie Sie die Konfigurationsdateien für Ihre Zwecke anpassen können.

Hinweis: Haben Sie alle Parameter während der Installationsvorganges richtig angegeben, brauchen sie keine Änderungen mehr an den Konfigurationsdateien vorzunehmen, um ein lauffähiges System zu erhalten. Nach jeder Änderung einer Konfigurationsdatei, ist ein neuer Start von rvs® Tiny notwendig.

3.1 Anpassen der globalen Konfigurationsdateien

Dateien

Es existieren zwei Konfigurationsdateien, die für die globalen Einstellungen von rvs® Tiny von besonderer Bedeutung sind.

- `$RVSTINY_HOME/conf/rvsTinyConfig.xml`
- `$RVSTINY_HOME/conf/rvsTiny.properties`

3.1.1 rvsTiny.properties

Diese Datei ist von der Art her eine Java-Properties-Datei, die Einträge der Form:

`<name>=<value>`

enthält.

Diese Datei verweist auf 3 wichtige Orte (beinhaltet drei wichtigen Parameter):

- das Installationverzeichnis von rvs® Tiny (Parameter: `RootDir`)
- die globale Konfigurationsdatei von rvs® Tiny (Parameter: `ConfigFile`)
- die globale Konfigurationsdatei für das System Logging (Parameter: `LogConfigFile`)

Im Folgenden sehen Sie eine Beispieldatei (mit # beginnende Kommentarzeilen werden ignoriert):

```
RootDir=C\:\\Programme\\rvsTiny
# configuration xml file
ConfigFile=C\:\\Programme\\rvsTiny\\conf\\rvsTinyConfig.xml
# log config file
LogConfigFile=C\:\\Programme\\rvsTiny\\conf\\rvsTinyLogger.xml
```


© gedas deutschland GmbH / Fachbereich rvs® Systems

3.1.2 rvsTinyConfig.xml

Elemente

In der folgenden Tabelle sind die wichtigsten Elemente aus der Datei `rvsTinyConfig.xml` aufgeführt.

ELEMENT	BESCHREIBUNG
DB	Verzeichnis für Jobverwaltung mit den Unterverzeichnissen ENDED, FAILED, RCV und SND. In den Unterverzeichnisse RCV und SND werden die temporären, nicht vollständig abgearbeiteten Jobs gespeichert. Ins Verzeichnis FAILED werden die misslungenen und ins Verzeichnis ENDED erfolgreich abgeschlossenen Jobs abgelegt. Diese Verzeichnisse sind auch in der rvs® Tiny GUI sichtbar (`Transfer-> Jobs`).
TEMP	Verzeichnis für temporäre Nutzung.
INBOX	Verzeichnis, in dem empfangene Dateien abgelegt werden.
OUTBOX	temporäres Verzeichnis für Versanddateien
ARCDIR	Archiv-Verzeichnis, in dem z.B. die Datei für das Revisionslog abgelegt wird.
LOGDIR	Verzeichnis für die beiden Log-Dateien: `monlog.log` und `tiny.log`.
JobstartConfigFile	Konfigurationsdatei für Jobstart.
StationsConfigFile	Stationskonfigurationsdatei: beinhaltet die Konfigurationsparameter der lokalen Station, der Nachbarstation und gerouteten Stationen.
HostAllowFile	Konfigurationdatei, die DNS-Namen oder IP-Adressen von Hosts enthält, von denen rvs® Tiny Kommandos an den rvs® Tiny Server geschickt werden dürfen.
HostDenyFile	Konfigurationdatei, die DNS-Namen oder IP-Adressen von Hosts enthält, von denen keine rvs® Tiny Kommandos an den rvs® Tiny Server geschickt werden dürfen.

Beispiel:

Auszug aus `rvsTinyConfig.xml`

```
...
<Environment>
<DB>c:\Programme\rvsTiny\jobs</DB>
<TEMP>c:\Programme\rvsTiny\files\temp</TEMP>
<INBOX>c:\Programme\rvsTiny\files\inbox</INBOX>
```

```
<OUTBOX>c:\Programme\rvsTiny\files\inbox</OUTBOX>
<ARCDIR>c:\Programme\rvsTiny\archive</ARCDIR>
<JobstartConfigFile>rvsTinyJobstart.xml</JobstartConfigFile>
<StationsConfigFile>rvsTinyStationlist.xml</StationsConfigFile>
<MonlogStylesheet>MonlogStylesheet.xslt</MonlogStylesheet>
<PersistenceArchive>RevisionLog.xml</PersistenceArchive>
<HostAllowFile>host.allow</HostAllowFile>
<HostDenyFile> host.deny</HostDenyFile>
<RMIServiceName>rvsTiny</RMIServiceName>
<RMIServiceHost>localhost</RMIServiceHost>
</Environment>
...
```

########## rvsTinyConfig.xml ####################

Sie können die Pfade für die Elemente DB, TEMP, INBOX, LOGDIR und ARCDIR selber festlegen und verändern.

Beispiel:

```
<DB>D:\rvsTiny\jobs</DB>
```

```
<TEMP>C:\Programme\rvsTiny\temp</TEMP>
```

Die Namen der Dateien rvsTinyJobstart und rvsTinyStationlist sind frei wählbar; sie müssen nur innerhalb des jeweiligen XML-Elements angegeben werden, gültige XML-Dateien sein und sich im conf-Verzeichnis befinden. Das gleiche gilt auch für HostAllowFile und HostDenyFile.

Beispiel:

```
<StationsConfigFile>stationen.xml</StationsConfigFile>
```

Die Datei stationen.xml ist eine Stationsliste im XML-Format mit den notwendigen rvs® Tiny-Parametern (siehe Kapitel 3.2) und befindet sich im conf Verzeichnis von rvs® Tiny.
Die Eintragungen für RMISeviceName und RMIServiceHost dienen für die interne Kommunikation und sollen nicht verändert werden.

Hinweis: Nach jeder Änderung einer Konfigurationsdatei müssen Sie rvs® Tiny stoppen und starten (siehe Kapitel 4.1 und 4.2).

3.2 Anpassen der Stationskonfiguration

Stationen

Die Konfiguration der Stationen ist auf zwei Wegen möglich:

– mittels der grafischen Benutzeroberfläche (GUI) oder

– mittels der XML-Stationskonfigurationsdatei.

Sie benötigen einen Eintrag in der Stationskonfiguration für Ihre lokale Station und für den Nachbarknoten (rvs® - Zentrale), mit dem Sie die Dateien austauschen. Die obligatorischen Parameter für die beiden

© gedas deutschland GmbH / Fachbereich rvs® Systems

Stationen (lokale und Nachbarknoten) wurden schon bei der Installation abgefragt und die Stationskonfigurationsdatei

`$RVSTINY_HOME/conf/rvsTinyStationlist.xml`

wurde entsprechend angepasst. Diese Stationen sind dann auch nach dem Start der rvs® Tiny GUI sichtbar. Wie Sie rvs® Tiny starten können, lesen Sie bitte im Kapitel 2.3. Wenn Sie keine weiteren gerouteten Stationen einrichten müssen, ist die Stationskonfiguration dadurch abgeschlossen und der Versand und Empfang von Dateien möglich.

3.2.1 Konfiguration von Stationen mittels GUI

Das Programm `startGUI` startet die rvs® Tiny GUI. Die Batchdatei `startGUI.bat` befindet sich im Verzeichnis `$RVSTINY_HOME\bin`.

Sie müssen sich in diesem Verzeichnis befinden, um das Programm ausführen zu können.

Beispiel:

```
startGUI
```

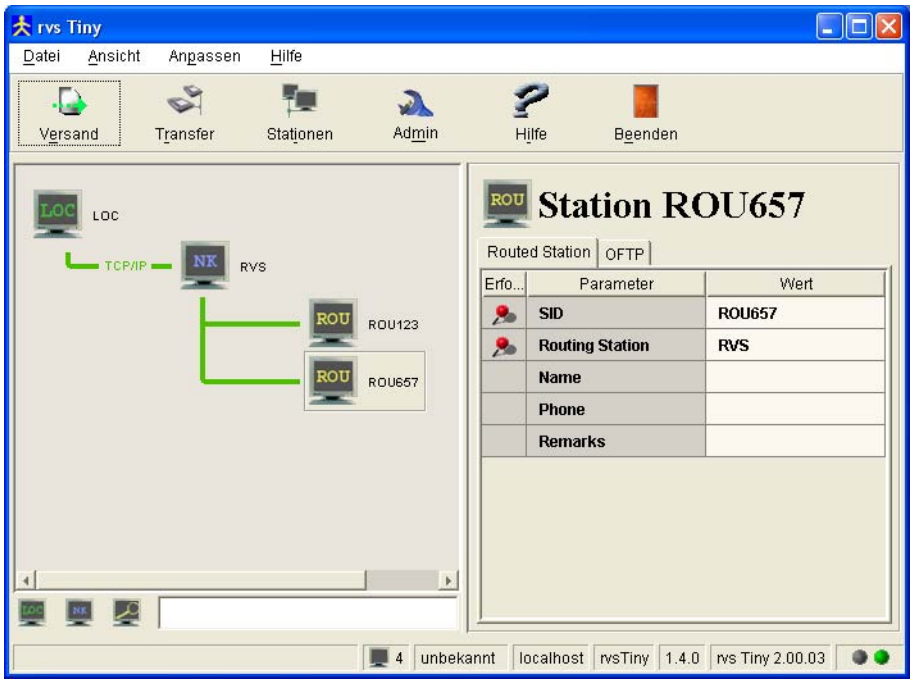
Dieser Befehl hat noch drei optionale Parameter. Mehr über die Syntax dieses Befehls lesen Sie bitte im Kapitel 2.5.

Die rvs® Tiny - Benutzeroberfläche können Sie auch mit Hilfe der rvs® Tiny - Programmgruppe starten (`Start -> Alle Programme -> rvs Tiny -> rvsTiny GUI`).

Nach dem erfolgreichen Start erscheint das folgende Fenster:

Im oberen Bereich des Fensters sehen Sie eine Menüzeile (Datei, Ansicht, Anpassen, Hilfe) und darunter eine Funktionsleiste, in der Sie über die einzelnen Symbole die wichtigsten Funktionen (Versand, Transfer, Stationen, Admin, Hilfe) aktivieren können.

Um zu den Stationen zu gelangen, öffnen Sie das Fenster Stationen in der Funktionsleiste.



Im linken Bereich des Stationsfensters sehen Sie den Stationsbaum, im rechten Teil eine Stationsparametertabelle.

Der Stationsbaum stellt alle in der rvs® Tiny-Datenbank existierenden Stationen (Ihre lokale, die rvs® - Zentrale und die gerouteten Partnerstationen) dar. Per Mausklick können Sie eine der dargestellten Stationen markieren.

In der Stationstabelle im rechten Fensterbereich werden Ihnen alle Parameter zur aktuell markierten Station angezeigt. Dabei können Sie mit Hilfe der verschiedenen Registerkarten unterschiedliche Parametergruppen konfigurieren.

Graue Felder deuten darauf hin, dass diese Parameter nicht editierbar sind.

Hinweis: Nachdem Sie einmal die Parameter einer Station gespeichert haben, können Sie nicht mehr die StationsID (SID) ändern. Die Änderung der StationsID würde bedeuten, dass Sie eine ganz neue Station konfigurieren möchten. Aus diesem Grunde sollten Sie zuerst die nicht mehr aktuelle Station löschen, um danach eine neue anzulegen.

© gedas deutschland GmbH / Fachbereich rvs® Systems

Eine Station löschen können Sie mit der Option **Station entfernen.** Zu dieser Option gelangen Sie, wenn Sie mit einem Klick der rechten Maustaste das Kontextmenü einer markierten Station öffnen.

Die Parameter, die für die Konfiguration einer Station obligatorisch sind, sind in der Spalte `Erfo...` (Erforderlich) mit dem Symbol gekennzeichnet.

Beispiel:

In der Registerkarte **OFTP** ist das `Odette Id`.

Lokale Station konfigurieren

Mögliche Registerkarten für die lokale rvs® Tiny - Station sind: Lokale Station, OFTP (Odette parameter), TCP/IP. Lesen Sie bitte die Kapitel 3.2.3 and 3.2.4 für eine detaillierte Erklärung der Stationsparameter. Die Parameter Name, Telefon, Bemerkung sind Kommentare. Die Registerkarte TCP/IP ist identisch mit dem XML-Element TCPIP_BASICS in der XML-Stationskonfigurationsdatei.

rvs® - Zentrale konfigurieren

Die möglichen Registerkarten für die Nachbarstation (rvs® - Zentrale) sind: Nachbarstation, OFTP, Leitungstyp, TCP/IP. Lesen Sie bitte in den Kapiteln 3.2.3 und 3.2.5 für eine detaillierte Erklärung der Stationsparameter.

Ein rechter Mauseklick auf die rvs® - Zentrale öffnet das Kontextmenü mit den Optionen: **geroutete Station hinzufügen** und **Verbindung aktivieren**. Mit der Option **Verbindung aktivieren** können Sie die rvs®-Zentrale aktivieren, um die Dateien, die für Ihre Station bestimmt sind, abzuholen. Mit der Option **geroutete Station hinzufügen** können Sie eine geroutete Station anlegen, die über die rvs® - Zentrale erreichbar ist.

Eine geroutete Station anlegen

Voraussetzung: Sie haben einen direkten Nachbarknoten (rvs®-Zentrale), über den Sie diese geroutete Station erreichen können, schon eingerichtet.

Sie klicken mit der rechten Maustaste auf die rvs® - Zentrale und wählen im Kontextmenü den Eintrag **geroutete Station hinzufügen**. Ein neuer Pfeil zeigt auf die neue Station. Für Sie ist es nicht von Beudeutung, mit welcher Verbindungsart diese Station erreicht wird (das regelt die rvs® - Zentrale). Deswegen ist auf diesem Pfeil auch keine Verbindungsart zu sehen.

Die möglichen Registerkarten für eine geroutete Station sind: Geroutete Station und OFTP. Lesen Sie bitte die Kapitel 3.2.3 and 3.2.6 für eine detaillierte Erklärung der Stationsparameter.

Eine geroutete Station kann nur über die rvs® - Zentrale erreicht werden. Mit dem Anlegen einer gerouteten Station ist diese Information (über

welche Nachbarstation) diese geroutete Station zu erreichen ist, im Parameter Routing Station hinterlegt. Dieser Parameter entspricht dem Parameter GATEWAY_STATION_NK im Element STATION_RT in der XML-Stationskonfigurationdatei.

Neben dem frei wählbaren Namen der Station (SID) ist nur noch die Odette ID in der Registerkarte OFTP einzutragen. Lesen Sie bitte in Kapitel 3.2.3 für eine detaillierte Erklärung der Odette ID.

Nachdem Sie einmal die Parameter einer Station gespeichert haben, können Sie nicht mehr den Namen (SID) ändern. Die Änderung des Namens würde bedeuten, dass Sie eine ganz neue Station konfigurieren möchten. Aus diesem Grunde sollten Sie zuerst die nicht mehr aktuelle Station löschen, um danach eine neue anzulegen.

Eine Station löschen können Sie mit der Option **Station entfernen.** Zu dieser Option gelangen Sie, indem Sie mit einem Klick der rechten Maustaste das Kontextmenü einer markierten Station öffnen.

3.2.2 Konfiguration von Stationen mittels der XML-Datei

Alternativ zur grafischen Oberfläche kann die Stationskonfiguration auch in der XML-Stationskonfigurationsdatei `rvsTinyStationlist.xml` durchgeführt werden. Das Element STATION_LOC aus der Stationskonfigurationsdatei ist äquivalent zu der Lokalen Station in der GUI, STATION_NK ist die Nachbar - Station und STATION_RT die Geroutete Station.

Auszug aus der Datei `rvsTinyStationlist.xml`:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<RVS_STATION_CONFIG>
<!-- Einstellungen für die Lokale Station -->
<STATION_LOC>
 . . .
</STATION_LOC>
<!--Einstellungen für die Nachbarstation -->
<STATION_NK>
 . . .
<STATION_NK>
<!----Einstellungen für geroutete Stationen  -->
<!--Für jede geroutete Station ein Eintrag -->
<STATION_RT>
. . .
</STATION_RT>
</RVS_STATION_CONFIG>
```

Kommentare

Bereiche, die mit dieser Zeichenfolge (<!--) beginnen und dieser Zeichenfolge (-->) enden, werden als Kommentar interpretiert.

Editieren Sie diese Datei, wenn Sie die obligatorischen Parameter (**ODETTE_ID**, **TCPIP_BASIC**) ändern müssen oder wenn Sie andere optionale Parameter wie z.B. **CONTACT** füllen möchten.

Benutzen Sie dazu ein Textbearbeitungsprogramm (z.B. `Edit`, `TextPad`). Achten Sie bitte bei der Bearbeitung der XML-Dateien

© gedas deutschland GmbH / Fachbereich rvs® Systems

darauf, dass Sie diese als gültiges XML-Dokument ablegen, sonst kann rvs® Tiny diese nicht lesen und eventuell nicht richtig starten.

Hinweis: Sie müssen dafür sorgen, dass die IP-Ports: RMI-Port (1099) und der Odette-Port (z.B. 3305) frei sind, damit die TCP/IP-Kommunikation einwandfrei funktioniert.

In den nächsten Kapiteln werden alle Stationsparameter ausführlich beschrieben.

3.2.3 Identifikation von rvs® Tiny Stationen

Stationsliste

In der Stationsliste gibt es mehrere Parameter, die rvs® Tiny Stationen identifizieren. Von diesen Parametern sind nur **SID**, **ODETTE_ID** und **TCPIP_BASIC** obligatorisch. Der Parameter SID wurde bei der Installation von rvs® Tiny vergeben; für die lokale Station lautet dieser Parameter LOC und für die rvs® - Zentrale RVS.

SID	Ist eine lokal eindeutige Stations-ID, die aus bis zu 16 Zeichen bestehen muss. Sie stellt eine streng lokale Definition dar; entfernte Stationen haben keinen Zugang zu diesem Namen, sie kennen nur die ODETTE-IDs. Bei der Wahl der Stations-IDs beachten Sie bitte, dass Sie diese IDs in allen Sendeeinträgen (SE) und allen residenten Einträgen (RE) benutzen müssen, um das Ziel und die Quelle der Übertragung zu identifizieren.
ODETTE_ID	ODETTE-ID ist eine weltweit eindeutige Identifikation aller Stationen, die das ODETTE Dateiübertragungsprotokoll (OFTP) verwenden. Der Name hat 25 Zeichen, die folgende Verteilung aufweisen: – der Buchstabe O, – eine aus 18 Zeichen bestehende Organisationsidentifikation, die von der ODETTE Kodifikationsgruppe bereitgestellt wird, und – eine aus 6 Zeichen bestehende Unteradresse, die von jeder Organisation selbst vergeben wird. Wenn Sie nur innerhalb Ihres geschlossenen Netzwerks kommunizieren, können Sie die Länge der ODETTE ID frei bestimmen, so dass sie in Ihrem Netzwerk eindeutig bleibt.
CONTACT	Kontaktinformationen, die im Folgenden noch weiter spezifiziert werden; diese Parameter sind optional.
../NAME	Name des Ansprechpartners, der für diese Station (Rechner) zuständig ist.
../ENTERPRISE	Firma / Unternehmen
../LOCATION	Ort

../STREET	Strasse
../STREETNUMBER	Hausnummer
../DEPARTMENT	Abteilung
../PHONE	Telefonnummer
../FAX	Faxnummer
../MAIL	E-Mail-Addresse
../INET	Webseite des Unternehmens
../FREETEXT	Zur freien Verfügung.
TCPIP_BASIC	Für den Verbindungsaufbau notwendigen TCP/IP-Parameter.
../IP_ADDR	IP-Adresse oder DNS-Name der eigenen oder der Partnerstation.
../PORT	Port auf dem ein TCP/IP-Listener gestartet werden soll; defaultmäßig 3305.
../SND_PW	Das Passwort, das rvs® Tiny der Nachbarstation sendet. Die ODETTE Passwörter werden immer zwischen den Nachbarstationen während des Aufbaus einer Session ausgetauscht und überprüft.
../REC_PW	Das Passwort, das rvs® Tiny von der Nachbarstation erwartet.
../EERP_OUT	Handhabung für das Senden von Empfangsbestätigungen (EERPs). NORMAL: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei und unmittelbares, aktives Versenden. HOLD: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei. Empfangsbestätigung wird jedoch erst nach Freigabe mit dem Programm handleEERP versendet. Default: NORMAL

Folgende Parameter sind zur Zeit in der Stationsliste aufgeführt, werden aber noch nicht ausgewertet:

– der Block **<TCPIP_REC>** mit den Parametern **<MAX_INCOMING_SESSIONS>** und **<MAX_OUTGOING_SESSIONS>**;
– der Parameter **ENABLED**.

Um sich in Deutschland für eine ODETTE ID anzumelden, wenden Sie sich bitte an:

ODETTE ID

Verband der Automobilindustrie e.V. (VDA)
Abt. Logistik
Postfach 17 05 63
60079 Frankfurt
Tel.: 069-7570-0

© gedas deutschland GmbH / Fachbereich rvs® Systems

Sie können eine vollständige OFTP Beschreibung unter folgender Adresse erhalten:

`http://www.odette.org/`

3.2.4 Stationseintrag STATION_LOC

lokale Station

Dieser Bereich enthält Informationen über die lokale Station, er darf nur genau einmal in der Stationskonfiguration vorhanden sein.

Beispiel für die lokale Station:

```
<STATION_LOC>
   <SID>LOC</SID>
   <ODETTE_ID>O0013005623GEDASSKK</ODETTE_ID>
   <CONTACT>
      <NAME>local station</NAME>
      <ENTERPRISE/>
      <LOCATION/>
      <STREET/>
      <STREET_NUMBER/>
      <DEPARTMENT/>
      <PHONE> </PHONE>
      <FAX/>
      <MAIL/>
      <INET/>
      <FREETEXT></FREETEXT>
   </CONTACT>
   <TCPIP_BASIC>
      <IP_ADDR>139.1.87.68</IP_ADDR>
      <PORT>3305</PORT>
   </TCPIP_BASIC>
   <TCPIP_REC>
      <MAX_INCOMING_SESSIONS>5</MAX_INCOMING_SESSIONS>
      <MAX_OUTGOING_SESSIONS>1</MAX_OUTGOING_SESSIONS>
   </TCPIP_REC>
   <ENABLED>Yes</ENABLED>
</STATION_LOC>
```

Im folgenden Beispiel ist die **SID** der lokalen Station `LOC`, die **ODETTE ID** `O0013005623GEDASSKK`. Diese Station ist für den Nachbarknoten unter der IP-Adresse `139.1.87.68` und dem Port `3305` erreichbar.

TCPIP_REC

Hinweis: Der Block **TCPIP_REC** mit den Parametern **MAX_INCOMING_SESSIONS** und **MAX_OUTGOING_SESSIONS** und der Parameter **ENABLED** werden zur Zeit noch nicht ausgewertet und sind für zukünftige Anwendungen gedacht. Diese Parameter sollen Sie nicht editieren oder löschen.

© gedas deutschland GmbH / Fachbereich rvs® Systems

3.2.5 Stationseintrag STATION_NK

rvs® - Zentrale

Dieser Bereich enthält Informationen nur über die eine direkte Nachbarstation, die rvs® - Zentrale. Auch dieser Bereich darf nur genau einmal in der Stationskonfiguration vorhanden sein.

```
<STATION_NK>
   <SID>RVS</SID>
   <ODETTE_ID>O0013005623GEDASMEL</ODETTE_ID>
   <CONTACT>
      <NAME>neighbour station</NAME>
      <ENTERPRISE/>
      <LOCATION/>
      <STREET/>
      <STREET_NUMBER/>
      <DEPARTMENT/>
      <PHONE> </PHONE>
      <FAX/>
      <MAIL/>
      <INET/>
      <FREETEXT></FREETEXT>
   </CONTACT>
   <LINE_TYPE>
      <LINE_SUSP>No</LINE_SUSP>
      <PSESSIONS>-1</PSESSIONS>
      <ACTIVE_CON_SETUP>Yes</ACTIVE_CON_SETUP>
      <DELAY>0</DELAY>
   </LINETYPE>
   <ODETTE>
      <SND_PW>H5C</SND_PW>
      <REC_PW>L3Y</REC_PW>
      <SND_BLOCKS>0</SND_BLOCKS>
      <REC_BLOCKS>0</REC_BLOCKS>
      <EX_BUF_CRE>0</EX_BUF_CRE>
      <EX_BUF_SIZ>0</EX_BUF_SIZ>
      <EERP_OUT>IMMEDIATE</EERP_OUT>
   </ODETTE>
   <TCPIP_BASIC>
      <IP_ADDR>139.1.65.42</IP_ADDR>
      <PORT>3305</PORT>
   </TCPIP_BASIC>
</STATION_NK>
```

SID	Stations-ID (siehe Kapitel 3.2.3).
ODETTE_ID	ODETTE-ID (siehe Kapitel 3.2.3).
CONTACT	Kontaktinformationen (siehe Kapitel 3.2.3).
LINE_TYPE	Für zukünftige Anwendungen, wird nicht ausgewertet.
ODETTE	Parameterblock, beinhaltet Parameter für OFTP.
../SND_PW	Das Passwort, das rvs® Tiny der Nachbarstation sendet. Die ODETTE Passwörter werden immer zwischen den Nachbarstationen während des Aufbaus einer Session ausgetauscht und überprüft.

../REC_PW	Das Passwort, das rvs® Tiny von der Nachbarstation erwartet.
../EX_BUF_CRE	Für zukünftige Anwendungen, wird nicht ausgewertet.
../EX_BUF_SIZ	Für zukünftige Anwendungen, wird nicht ausgewertet.
../EERP_OUT	Handhabung für das Senden von Empfangsbestätigungen (EERPs). NORMAL: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei und unmittelbares, aktives Versenden. HOLD: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei. Empfangsbestätigung wird jedoch erst nach Freigabe mit dem Programm handleEERP versendet. Default: NORMAL
TCPIP_BASIC	Für den Verbindungsaufbau notwendigen TCP/IP-Parameter.
../IP_ADDR	IP-Adresse oder DNS-Name der eigenen oder der Partnerstation.
../PORT	Port auf dem ein TCP/IP-Listener gestartet werden soll; defaultmäßig 3305.

3.2.6 Stationseintrag STATION_RT

Routing-Station

Dieser Eintrag definiert für jede rvs® Station, durch welche Nachbarstationen sie erreicht werden kann.

```
<STATION_RT>
   <SID>ROUTEDSTATION</SID>
   <ODETTE_ID>OROUT</ODETTE_ID>
   <GATEWAY_STATION_NK>RVS</GATEWAY_STATION_NK>
   <CONTACT>
      <NAME/>
      <ENTERPRISE/>
      <LOCATION/>
      <STREET/>
      <STREET_NUMBER/>
      <DEPARTMENT/>
      <PHONE/>
      <FAX/>
      <MAIL/>
      <INET/>
      <FREETEXT/>
   </CONTACT>
   <ODETTE>
      <EERP_OUT>IMMEDIATE</EERP_OUT>
   </ODETTE>
</STATION_RT>
```

SID	Stations-ID (siehe Kapitel 3.2.3).
ODETTE_ID	ODETTE-ID (siehe Kapitel 3.2.3).
CONTACT	Kontaktinformationen (siehe Kapitel 3.2.3).

© gedas deutschland GmbH / Fachbereich rvs® Systems

GATEWAY_STATION_NK	Dieses Feld enthält die SID der Station über die die Kommunikation mit der hier konfigurierten gerouteten Station ablaufen soll. Dies kann bei rvs® Tiny nur die direkte Nachbarstation (rvs® - Zentrale) sein , da rvs® Tiny nur eine direkte Nachbarstation erlaubt.
ODETTE	Parameterblock, beinhaltet Parameter für OFTP.
../EERP_OUT	Handhabung für das Senden von Empfangsbestätigungen (EERPs). NORMAL: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei und unmittelbares, aktives Versenden. HOLD: Generieren einer Empfangsbestätigung nach erfolgreichem Empfang einer Datei. Empfangsbestätigung wird jedoch erst nach Freigabe mit dem Programm handleEERP versendet. Default: NORMAL
TCPIP_BASIC	Für den Verbindungsaufbau notwendigen TCP/IP-Parameter.
../IP_ADDR	IP-Adresse oder DNS-Name der eigenen oder der Partnerstation.
../PORT	Port auf dem ein TCP/IP-Listener gestartet werden soll; defaultmäßig 3305.

3.3 Konfiguration der JobStarts

Die JobStarts beinhalten Regeln, die dazu führen, dass bei entsprechenden ein- bzw. ausgehenden Dateien spezielle Programme gestartet werden können. Sie entsprechen den Residenten Empfangseinträgen (Jobstarts nach Empfang) und den Jobstarts nach Sendeversuchen in rvs®.

JobFilter

Falls mehr als ein JobFilter auf den Sende- bzw. Empfangsjob zutrifft, so werden alle Programme der zutreffenden JobFilter gestartet, wobei die Reihenfolge nicht festgelegt werden kann.

Die Jobstarts können über die rvs® Tiny GUI oder in der XML-Jobstarts-Konfigurationsdatei `rvsTinyJobstart.xml` konfiguriert werden.

Konfiguration der Jobstarts mittels GUI

Die Jobstarts werden im Administrationsfenster konfiguriert. Zu diesem Fenster gelangen Sie, indem Sie das Symbol Admin in der Funktionsleiste anklicken.

Wenn Sie sich im Admin-Fenster befinden, können Sie zwischen Jobstarts nach Empfang und Jobstarts nach Sendeversuch wählen, je nachdem, ob Sie eine bestimme Aktion gekoppelt an den Empfang oder an den Versand Dateien auslösen möchten.

Einen neuen Jobstart können Sie anlegen, indem Sie im Admin-Baum `Jobstarts` markieren und mit der rechten Maustaste den Menüpunkt

Neuen Eintrag erstellen wählen. Sie können dann im neu erstellten Eintrag bei der Richtung festlegen, ob dies ein Jobstart nach Empfang oder ein Jobstart nach Sendeversuch werden soll. Die zweite Möglichkeit ist, sich sofort für die Richtung zu entscheiden, indem Sie Jobstart nach Empfang oder Jobstart nach Sendeversuch markieren und anschließend mit der rechten Maustaste den Menüpunkt **Einen neuen Eintrag erstellen** wählen.

Nachdem Sie einen neuen Jobstarteintrag erstellt haben, bieten sich in der Jobstartmaske folgende Filter an: Richtung (Direction), SID, VDSN, Prozess und Sendversuche (nur beim Jobstart nach Sendeversuch). Lesen Sie bitte detaillierter zu den Filtern in der Tabelle im nächsten Abschnitt.

Konfiguration der Jobstarts in der XML-Datei

Die XML-Konfigurationsdatei `rvsTinyJobstart.xml` dient der Konfiguration der Jobstarts.

Diese Datei ist wie alle anderen rvs® Tiny - Konfigurationsdateien auch im XML-Format .

```
<jobstarterData>
       <jobfilters>
              <jobfilter>
                     <vdsn></vdsn>
                     <sid></sid>
                     <direction>SND</direction>
                     <sendAttempts>0</sendAttempts>
                     <process>C:\jobstart.bat</process>
              </jobfilter>
              <jobfilter>
               ...
              </jobfilter>
              ...
        </jobfilters>
</jobstarterData>
```

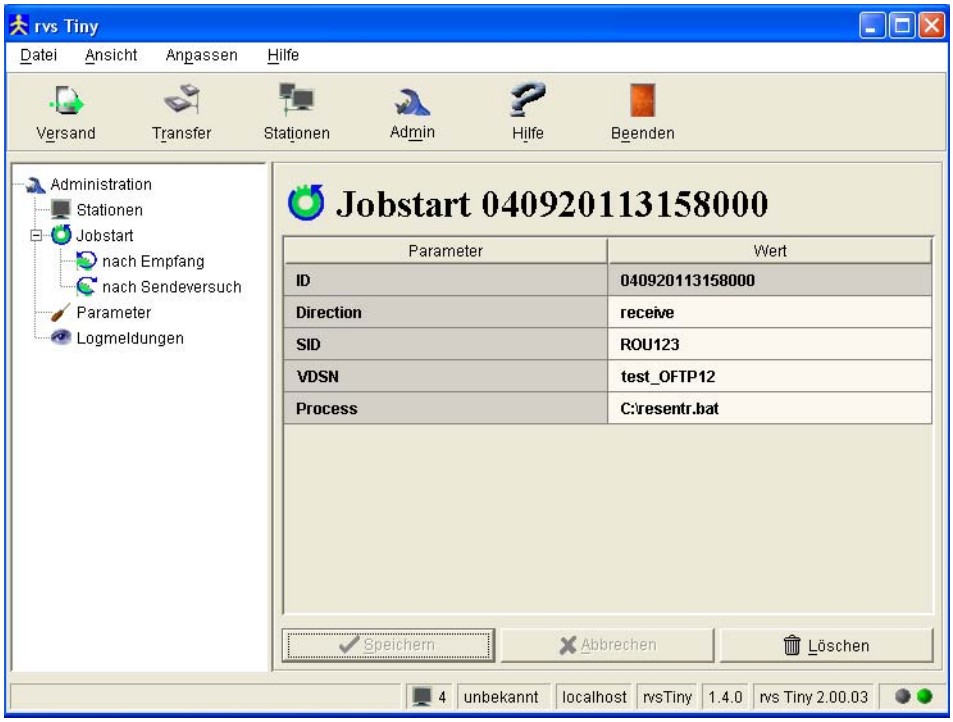
Diese Datei beinhaltet beliebig viele jobFilter-Elemente. Die einzelnen Unterelemente von JobFilter werden in der folgenden Tabelle genau beschrieben.

vdsn	Virtueller Dateiname (Regulärer Ausdruck als Filter).
sid	Stations-Id (Regulärer Ausdruck als Filter für die StationsId)
direction	Legt Filterbedingung für die Richtung der Kommunikation fest. Mögliche Werte: SND (Bei Dateiversand), RCV (Bei Dateiempfang).
sendAttempts	Anzahl der fehlgeschlagenen Sendeversuche. Wird hier "0" eingetragen, so bedeutet dies erfolgreicher Dateiversand.

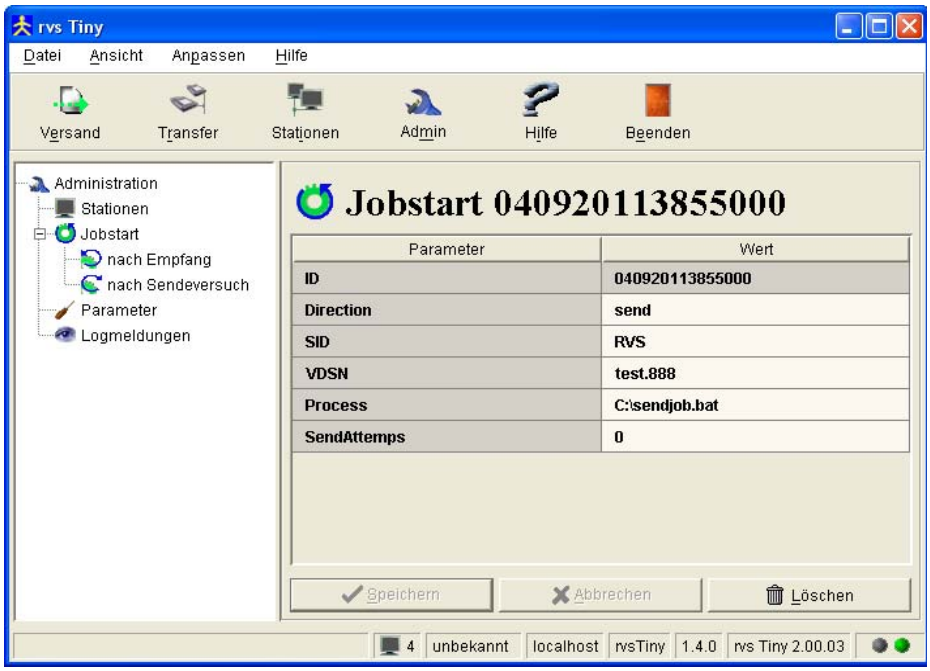
© gedas deutschland GmbH / Fachbereich rvs® Systems

process (mandatory)	Programm, das gestartet werden soll, wenn die Filterbedingungen alle zutreffen. Den Programmen wird ein fest definierter Satz von Parametern übergeben. Parameter: 1. jobId 2. Stations ID (des Senders bzw. des Empfängers) 3. Dateiname der gesendeten/bzw. empf. Datei 4. VDSN 5. Datum und Uhrzeit des Jobs 6. Anzahl der Sendeversuche.

Beispiel (GUI) Beim Empfang der Datei mit dem virtuellen Dateinamen `test_OFTP12` von der gerouteten Station `ROU123`, startet rvs® Tiny das Programm `C:\resentr.bat`.



Beispiel (GUI) Wenn rvs® Tiny eine Datei mit dem Namen `test.888` an die rvs® - Zentrale RVS erfolgreich (SendAttempts=0) versendet, startet das Programm `C:\sendjob`.



Jobstarteintrag löschen

Wenn Sie einen Jobstarteintrag löschen möchten, müssen Sie ihn zuerst mit Doppelklick auswählen. Im neuen Fenster haben Sie dann die Möglicheit, den ausgewählten Jobstart mit der Schaltfläche **Löschen** zu entfernen.

© gedas deutschland GmbH / Fachbereich rvs® Systems

4 Arbeiten mit rvs® Tiny

Batch-Dateien

In diesem Kapitel werden alle Programme, die Ihnen für den täglichen Umgang mit rvs® Tiny zur Verfügung stehen, beschrieben. Diese Programme befinden sich als Batch-Dateien im Verzeichnis `$RVSTINY_HOME\bin`.

Hinweis: Sie müssen sich im Verzeichnis `$RVSTINY_HOME\bin` befinden, wenn Sie ein rvs® Tiny Programm aufrufen.

Manche von den rvs® Tiny - Programmen (wie z.B. GUI, Start und Stopp von rvs® Tiny) stehen auch über die rvs® Tiny - Programmgruppe zur Verfügung (`Start -> Alle Programme -> rvsTiny`).

4.1 Starten des rvs® Tiny Servers

`startServer`

Mit dem Programm `startServer` können Sie rvs® Tiny starten.

Beispiel:

```
startServer
```

Es können keine Parameter übergeben werden. Ein erfolgreicher Start sieht folgendermaßen aus:

```
*
* rvs Tiny Server has started.
*
```

Hinweis: Es ist auch möglich, den rvs® Tiny Server über die rvs® Tiny - Programmgruppe zu starten (`Start -> Programme -> rvsTiny -> rvsTiny Start`).

4.2 Stoppen des rvs® Tiny Servers

`stopServer`

Stoppen Sie rvs® Tiny mit dem Programm `stopServer`.

Syntax:

```
stopServer -m <mode> [-verbose]
```

Alle Parameter sind optional:

-m <mode>	Zeit für Beendigung der Jobs, bevor rvs® Tiny gestoppt wird. Mögliche Werte: 0 (default; 120 Sekunden), 1 (60 Sekunden), 2 (30 Sekunden), 3 (20 Sekunden), 4 (10 Sekunden).
-verbose	Ausgabe von ausführlichen Meldungen.
-help	Fordert Hilfe (Usage) an.
-?	Fordert Hilfe (Usage) an.

Beispiel:

```
stopServer
```

Ergebnis: Der Server wird in 120 Sekunden gestoppt.

Beispiel:

```
stopServer -m 3
```

Ergebnis: Der Server wird in 20 Sekunden gestoppt.

```
*
* rvs Tiny Server has stopped.
*
```

Hinweis: Es ist auch möglich, den rvs® Tiny Server über die rvs® Tiny - Programmgruppe zu stoppen (`Start -> Programme -> rvsTiny -> rvsTiny Stop`).

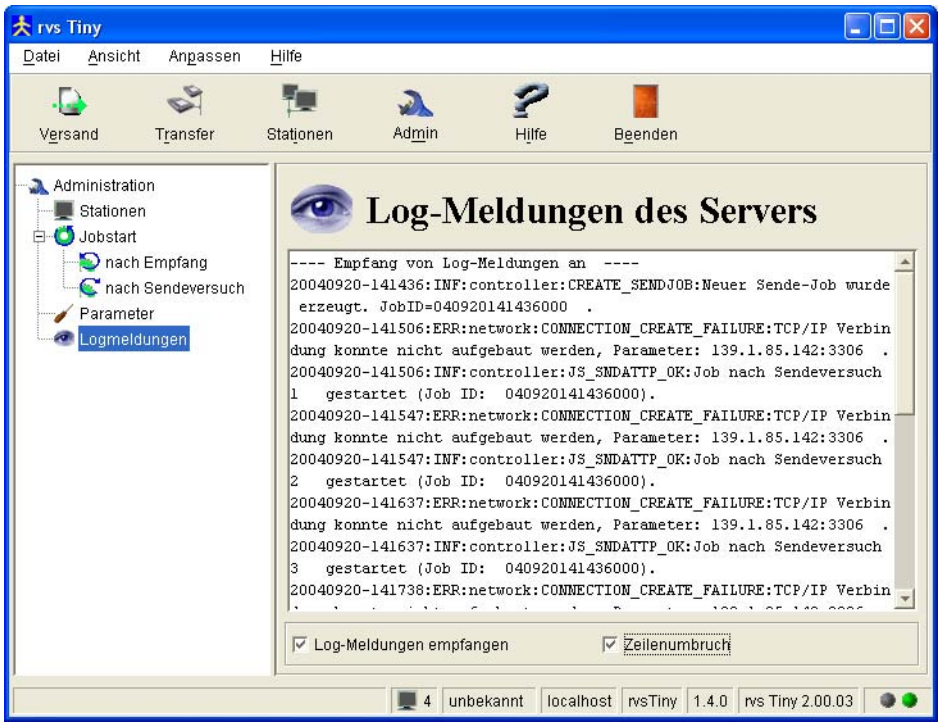
4.3 Meldungen des Monitors anzeigen

Die Meldungen des Monitors können Sie über die GUI oder mit dem Batch-Programm `showMonitorLog` anzeigen lassen.

GUI

Die Meldungen des Monitors können im Admin-Fenster im Baum Administration, Menüpunkt Log Messages angezeigt werden. Zum Admin-Fenster gelangen Sie, wenn Sie das Symbol Admin in der Funktionsleiste anklicken. Wie Sie die rvs® Tiny GUI starten können, lesen Sie bitte in Kapitel 2.5.

© gedas deutschland GmbH / Fachbereich rvs® Systems



Eingabeaufforderung (Kommandozeile)

Benutzen Sie das Programm `showMonitorLog` um die aktuellen Meldungen des Monitors zu verfolgen und eventuell Fehlermeldungen zu analysieren.

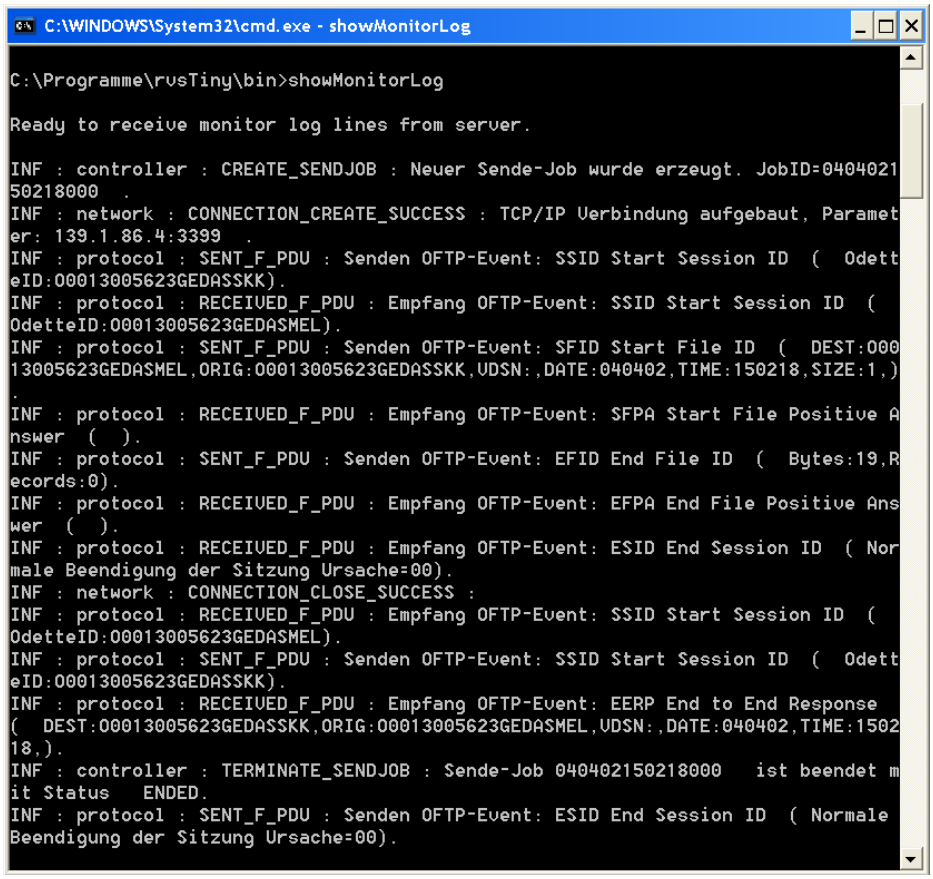
Syntax:

```
showMonitorLog [-verbose]
```

Optionale Parameter:

-verbose	Ausgabe von ausführlichen Meldungen.
-help	Gibt Beschreibung zum aktuellen Kommando aus
-?	Fordert Hilfe an.

Beispiel:


C:\WINDOWS\System32\cmd.exe - showMonitorLog
C:\Programme\rvsTiny\bin>showMonitorLog
Ready to receive monitor log lines from server.
INF : controller : CREATE_SENDJOB : Neuer Sende-Job wurde erzeugt. JobID=040402150218000 .
INF : network : CONNECTION_CREATE_SUCCESS : TCP/IP Verbindung aufgebaut, Parameter: 139.1.86.4:3399 .
INF : protocol : SENT_F_PDU : Senden OFTP-Event: SSID Start Session ID (OdetteID:O0013005623GEDA$SKK).
INF : protocol : RECEIVED_F_PDU : Empfang OFTP-Event: SSID Start Session ID (OdetteID:O0013005623GEDA$MEL).
INF : protocol : SENT_F_PDU : Senden OFTP-Event: SFID Start File ID (DEST:O0013005623GEDA$MEL,ORIG:O0013005623GEDA$SKK,VDSN:,DATE:040402,TIME:150218,SIZE:1,).
INF : protocol : RECEIVED_F_PDU : Empfang OFTP-Event: SFPA Start File Positive Answer ( ).
INF : protocol : SENT_F_PDU : Senden OFTP-Event: EFID End File ID (Bytes:19,Records:0).
INF : protocol : RECEIVED_F_PDU : Empfang OFTP-Event: EFPA End File Positive Answer ( ).
INF : protocol : RECEIVED_F_PDU : Empfang OFTP-Event: ESID End Session ID (Normale Beendigung der Sitzung Ursache=00).
INF : network : CONNECTION_CLOSE_SUCCESS :
INF : protocol : RECEIVED_F_PDU : Empfang OFTP-Event: SSID Start Session ID (OdetteID:O0013005623GEDA$MEL).
INF : protocol : SENT_F_PDU : Senden OFTP-Event: SSID Start Session ID (OdetteID:O0013005623GEDA$SKK).
INF : protocol : RECEIVED_F_PDU : Empfang OFTP-Event: EERP End to End Response (DEST:O0013005623GEDA$SKK,ORIG:O0013005623GEDA$MEL,VDSN:,DATE:040402,TIME:150218,).
INF : controller : TERMINATE_SENDJOB : Sende-Job 040402150218000 ist beendet mit Status ENDED.
INF : protocol : SENT_F_PDU : Senden OFTP-Event: ESID End Session ID (Normale Beendigung der Sitzung Ursache=00).

Die Meldungen des Monitors werden in die Datei $RVSTINY_HOME `\rvsTiny\log\monlog.log` geschrieben.

4.4 Aktivieren einer Station

`activate Station`

Mit dem Programm `activateStation` können Sie die rvs® - Zentrale aktivieren, um die Dateien, die für Ihre Station bestimmt sind, abzuholen (siehe auch Kapitel 3.2.1).

Syntax:

```
activateStation [-verbose]
```

Optionale Parameter:

-verbose	Ausgabe von ausführlichen Meldungen.
-help	Gibt Beschreibung zum aktuellen Kommando aus.
-?	Fordert Hilfe an.

Wenn die Aktivierung der rvs® - Zentrale nicht erfolgreich sein sollte (wie z.B. wegen einer falschen IP-Adresse), müssen Sie auf die Nachrichten

© gedas deutschland GmbH / Fachbereich rvs® Systems

in dem Eingabeaufforderungsfenster, die mit Meldung oder Error beginnen, achten.

Im Falle einer falschen IP-Adresse, muss die Konfigurationsdatei mit der Stationsliste korrigiert werden. Nach der Speicherung der Konfigurationsdatei ist ein erneutes Stoppen und Starten von rvs® Tiny, notwendig.

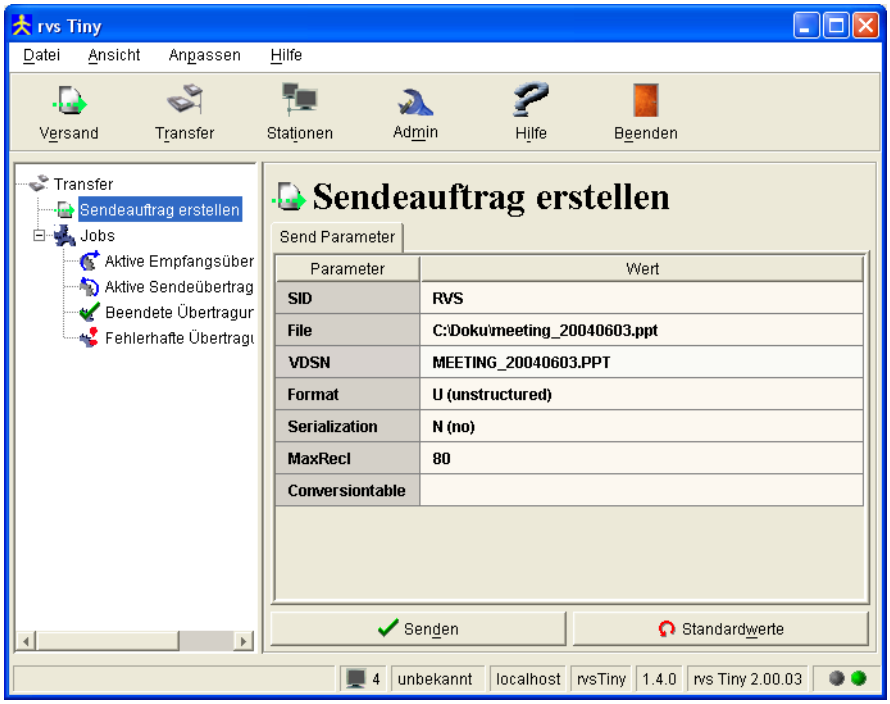
4.5 Versand einer Datei

`createSendJob`

Der Versand einer Datei ist mittels der GUI und mittels der Eingabeaufforderung möglich.

GUI

Wenn Sie das Symbol Versand oder Transfer in der Funktionsleiste anklicken, gelangen Sie ins Transfer-Fenster, wo Sie im Transfer-Baum den Menüpunkt **Sendeauftrag erstellen** wählen können. Im Fenster „Sendeauftrag erstellen“ können Sie Ihre Daten für den aktuellen Sendeauftrag eintragen oder auswählen.



Pflicht-Parameter:

SID	StationsID des Empfängers
File (Datei)	Dateiname der zu versendenden Datei

VDSN	Virtueller Dateiname; das ist der Dateiname, der für die ODETTE-Übertragung benutzt wird, darf maximal 26 Zeichen lang sein.
Optionale Parameter:	
Format	Format der zu übertragenden Datei – T (Text): eine Folge von ASCII-Zeichen – F (Fest): feste Satzlänge – V (Variabel): variable Satzlänge – U (Unstrukuriert): binäre Datei Die Dateien im Format F oder V müssen Textdateien mit der gewünschten Satzlänge (ohne CR/LF) sein. **Beispiel**: Wenn Sie eine Datei im Format Fixed mit der Satzlänge 80 versenden möchten, muss diese Datei eine Textdatei sein, in welcher jede Zeile die Länge 80 ohne CR/LF hat. Dann sind die folgenden Parameter zu setzen: Format=F, MaxRecl=80.
Serialization	**Y**(Yes)/**N**(No) Diese Option gewährleistet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien, die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Label) haben. In der GUI dient VDSN als Label.
MaxRecl	Für Dateien im Format Fest geben Sie die feste Satzlänge an, mit der der Empfänger die Datei interpretieren soll. Dies ist die Satzlänge jedes Satzes bis zum Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen). Für Dateien im Format Variabel geben Sie die maximale Satzlänge an. Siehe Beispiel für Parameter Format.

© gedas deutschland GmbH / Fachbereich rvs® Systems

Conversiontable	Für ASCII(ANSI) - EBCDIC Umwandlung stehen folgende Konvertierungstabellen zur Verfügung: ASCII-IBM037, ASCII-IBM237, ANSI-IBM037, ANSI-IBM273. Für EBCDIC - ASCII(ANSI) Umwandlung stehen folgende Konvertierungstabelle zur Verfügung: IBM037-ASCII, IBM237-ASCII, IBM037-ANSI, IBM273-ANSI.
-h	Hilfe
-verbose	ausführliche Meldungen
-?	Hilfe

Eingabeaufforderung

Mit dem Programm `createSendJob` erstellen Sie einen Sendeauftrag.

Syntax:

```
createSendJob -d <filename> -s <receiver sid>
[-I <input code> -O <output code>
-t <table name>]
[-F <format>] [-M <length>]
[-S <serialize> -l <label>]
[-v <vdsn>] [-h?][-verbose]
```

Benötigte Parameter:

-d <filename>	Dateiname der zu versendenden Datei.
-s <receiver sid>	StationsID des Empfänger.
-v <vdsn>	Virtueller Dateiname; das ist der Dateiname, der für die ODETTE-Übertragung benutzt wird, darf maximal 26 Zeichen lang sein.

Optionale Sende-Parameter:

-F <format>

Format der zu übertragenden Datei

- T (Text): eine Folge von ASCII-Zeichen
- F (Fest): feste Satzlänge
- V (Variabel): variable Satzlänge
- U (Unstrukuriert): binäre Datei

Die Dateien im Format F oder V müssen Textdateien mit der gewünschten Satzlänge (ohne CR/LF) sein.

Beispiel: Wenn Sie eine Datei im Format Fixed mit der Satzlänge 80 versenden möchten, muss diese Datei eine Textdatei sein, in welcher jede Zeile die Länge 80 ohne CR/LF hat. Dann sind die folgenden Parameter zu setzen: `-F=F`, `-M=80`.

-m <length>

Für Dateien im Format Fest geben Sie die feste Satzlänge an, mit der der Empfänger die Datei interpretieren soll. Dies ist die Satzlänge jedes Satzes bis zum Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen). Für Dateien im Format Variabel geben Sie die maximale Satzlänge an. Siehe Beispiel für Parameter Format.

-I <input code>

Für ASCII(ANSI) - EBCDIC Umwandlung stehen folgende Konvertierungstabellen zur Verfügung: ASCII-IBM037, ASCII-IBM237, ANSI-IBM037, ANSI-IBM273.

Für EBCDIC - ASCII(ANSI) Umwandlung stehen folgende Konvertierungstabelle zur Verfügung:
IBM037-ASCII, IBM237-ASCII, IBM037-ANSI, IBM273-ANSI.

Beispiel: Wenn Sie die Umwandlungstabelle IBM237-ASCII benutzen möchten, geben Sie als Input Code IBM237 (`-I IBM237`) und als Output Code ASCII (`-O ASCII`) an.

-O <output code>

siehe Input Code.

-t <table name>

Pfad und Name Ihrer eigenen Codeumwandlungstabelle. Lesen Sie bitte das rvsXP-Benutzerhandbuch, Kapitel „Codeumwandlung“ für mehr Informationen.

© gedas deutschland GmbH / Fachbereich rvs® Systems

-S <serialize>	**Y**(Yes)/**N**(No). Diese Option gewährleistet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien, die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Label) haben. In der GUI dient VDSN als Label und hier (in der Eingabeaufforderung) kann Label (siehe Parameter Label) angegeben werden.
-l <label>	Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (Label) haben. Siehe auch Option `-s <serialize>`.
-h	Fordert Hilfe (Usage) an.
-verbose	Ausgabe von ausführlichen Meldungen.
-?	Fordert Hilfe (Usage) an.

Beispiele:

```
createSendJob -d C:\text.txt -s RVS -v test
```

In diesem Beispiel wird die Datei `C:\text.txt` mit dem virtuellen Namen `test` an die rvs® - Zentrale RVS gesendet.

```
createSendJob -d C:\text.txt -s RVS -v OFTP_TEST
-F F -M 80 -I ANSI -O IBM273
```

In diesem Beispiel wird die Datei `C:\test.txt` an die rvs® - Zentrale RVS mit dem virtuellen Namen `OFTP_TEST` gesendet; diese Datei ist eine Textdatei, in welcher jede Zeile die Länge 80 (`-F F -M 80`) ohne CR/LF hat. Vor der Übertragung findet noch eine Codeumwandlung (ANSI-IBM273) statt.

```
createSendJob -d C:\part.txt -s RVS -v PART
 -S Y -l AUTO
```

In diesem Beispiel wird die Datei `C:\part.txt` an die rvs® - Zentrale RVS mit dem virtuellen Namen PART gesendet; diese Datei gehört zu der Gruppe der serialisierten Dateien mit dem Label `AUTO`.

Wenn ein Sendeauftrag erfolgreich generiert wurde, bekommen Sie die Meldung: `createSendJob exited with return code 0`.

Während der Sendeauftrag in Bearbeitung ist, entsteht im Verzeichnis SND eine temporäre Datei mit einem Namen wie z.B. `040329170027000`. Dieser Name setzt sich aus Datum (`040329`), Uhrzeit (`170027`) und dreistelliger laufender Nummerierung (`000`) für Dateien, die in der gleichen Sekunde ankommen, zusammen.

Im folgenden Beispiel wartet der Job `040329170027000` auf eine Empfangsbestätigung (**EERP**) von der rvs® - Zentrale und aus diesem Grunde befindet er sich noch im SND-Verzeichnis. SND und RCV sind

Verzeichnisse, in denen die Jobs, die gerade bearbeitet werden, temporär abgelegt werden. Sie können im Feld Status lesen, im welcher Bearbeitungsphase sich der Job gerade befindet. WF_EERP (**W**aiting **F**or **E**ERP) bedeutet: warten auf EERP (Empfangsbestätigung).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<Job>
       <ID>040329170027000</ID>
       <Static>
              <FileName>C:\Answer.txt</FileName>
              <VDSN>TESTDATEI</VDSN>
              <SID>RVS</SID>
              <Direction>SND</Direction>
              <Date>040329</Date>
              <Time>170027</Time>
       </Static>
       <Dynamic>
              <RestartPos>0</RestartPos>
              <RecCount>0</RecCount>
              <FilePos>19</FilePos>
                 <RcvBytes>0</RcvBytes>
                 <SendAttempts>0</SendAttempts>
                 <Status>WF_EERP</Status>
          </Dynamic>
</Job>
```

Wenn ein Job bis zum Ende erfolgreich abgearbeitet wurde (z.B. EERP ist angekommen), kopiert rvs® Tiny ihn ins Verzeichnis ENDED. Das Verzeichnis FAILED ist für die Jobs, die nicht erfolgreich abgearbeitet werden konnten.

Einige mögliche Statuswerte für Sendjobs:

- RESTART=1 (Warte nach Fehler, um die Datei wieder zu senden)
- WF_SFID_ANSWER=2 (Warte auf SFID Antwort für bereits gesendete SFID)
- WF_CDT=3 (warte auf Credit)
- WF_EFID_ANSWER=4 (warte auf Antwort auf bereits gesendete EFID)
- WF_EERP=5 (Warte auf EERP)
- ENDED=6 (Sendjob ist beendet).
- ENDED_WITH_JS_ERROR=7 (Sendjob endete mit Fehler während des Aufrufs von Jobstart).
- FATAL_ERROR=8 (Schwerwiegender Fehler des Sendjobs)

Einige mögliche Statuswerte von Empfangjobs:

- RESTART (Warte nach dem Fehler, um die Datei wieder zu empfangen).
- RESTART_AFTER_EFNA=2 (Die Datei konnte nicht empfangen werden, EFNA ist vielleicht gesendet, der Partner wird wahrscheinlich die Datei wieder senden).

© gedas deutschland GmbH / Fachbereich rvs® Systems

- RESTART_AFTER_EFPA_FAILURE=3 (Die Datei wurde empfangen, aber EFPA war nicht erfolgreich. Der Partner wird wahrscheinlich die Datei erneut senden.)
- RECEIVING=4 (Empfang der Daten nach EFPA)
- EERP_HOLDED=5 (Die Datei wurde vollständig empfangen. EERP ist im HOLD-Status und soll freigegeben werden).
- EERP_RELEASED=6 (Der Benutzer hat EERP freigegeben, aber noch nicht versendet).
- EERP_DELETED=7 (Der Benutzer hat EERP gelöscht. Der Job wird gestoppt).
- ENDED=8 (Empfangsjob wurde erfolgreich beendet nach dem Versand von EERP).
- ENDED_WITH_JS_ERROR=9 (Receivejob endete mit Fehler während des Aufrufs von Jobstart).
- FATAL_ERROR=10 (Schwerwiegender Fehler des Empfangsjobs)

Hinweis: Lesen Sie bitte im rvs® portable, Referenzhandbuch (Kapitel 3.3) mehr zu dem ODETTE Protokoll (z.B. über EFNA, SFID, EFID, EERP,...).

4.6 Synchronisation von Sendeaufträgen

Der Dateitransfer über OFTP wird asynchron abgearbeitet. Bei Erstellung eines Sendeauftrags wird eine Datei lediglich zum Senden bereitgestellt. Der eigentliche Versand erfolgt, sobald rvs® eine Verbindung zur Partnerstation aufbauen konnte. Gerade im automatisierten Berieb ist es jedoch oft erforderlich, direkt zu reagieren, wenn eine Datei innerhalb einer bestimmten Zeit nicht versendet werden konnte, oder der Versand erfolgreich war.

rvs® Tiny stellt eine Methode zur Verfügung, mit der der Dateiversand quasi synchron abgewickelt werden kann. Diese Methode kehrt erst zum Anrufer zurück, wenn der Versand erfolgreich war oder ein Fehler auftrat. Es kann die Anzahl der Sendeversuche oder die Zeit angegeben werden, innerhalb der die erfolgreiche Abwicklung des Dateiversandes erwartet wird. Wird dieser Rahmen überschritten, gilt der Versand als fehlerhaft. Der Versand gilt als erfolgreich, wenn eine OFTP-Quittung (EERP) empfangen wurde.

Mit dieser Methode kann außerdem vor dem Dateiversand eine Konvertierung einer EDI Nachricht mit dem EDI-Konverter WEDIConv durchgeführt werden.

Das Programm `convertAndSend` bietet Ihnen diese Funktionalität. Dieses Programm ist dem Programm `createSendJob` ähnlich. Es hat zusätzlich die Funktionalität zum synchronisierten Dateiversand und für die Konvertierung der EDI-Nachricht mit dem EDI Konverter **WEDIConv**. Die Konvertierung kann in zwei Schritten erfolgen. Dem ersten Schritt kann hierbei ein XSLT - Stylesheet mitgegeben werden, mit dem das Ergebnis von dem XML-Format nochmals transformiert wird. Lesen Sie bitte das WEDIConv Benutzerhandbuch, um mehr Informationen über die Konvertierung der EDI-Nachrichten zu erhalten.

Syntax:

```
convertAndSend -d <filename> -s <receiver sid>
-v <vdsn> [-I <input code> -O <output code>
-t <table name>][-F <format>] [-M <length>]
[-S <serialize> -l <label>][-za <attempts>]
[-zt <timeout>][converter parms] [-h?][-verbose]
```

Benötigte Parameter:

-d <filename>	Dateiname der zu versendenden Datei.
-s <receiver sid>	StationsID des Empfängers.
-v <vdsn>	Virtueller Dateiname; das ist der Dateiname, der für die ODETTE-Übertragung benutzt wird, darf maximal 26 Zeichen lang sein.

© gedas deutschland GmbH / Fachbereich rvs® Systems

Optionale Sende-Parameter:

-F <format> Format der zu übertragenden Datei

- T (Text): eine Folge von ASCII-Zeichen
- F (Fest): feste Satzlänge
- V (Variabel): variable Satzlänge
- U (Unstrukturiert): binäre Datei

Die Dateien im Format F oder V müssen Textdateien mit der gewünschten Satzlänge (ohne CR/LF) sein.

Beispiel: Wenn Sie eine Datei im Format Fixed mit der Satzlänge 80 versenden möchten, muss diese Datei eine Textdatei sein, in welcher jede Zeile die Länge 80 ohne CR/LF hat. Dann sind die folgenden Parameter zu setzen: `-F=F`, `-M=80`.

-m <length> Für Dateien im Format Fest geben Sie die feste Satzlänge an, mit der der Empfänger die Datei interpretieren soll. Dies ist die Satzlänge jedes Satzes bis zum Zeilenwechsel (CR/LF bei MS Windows, LF bei UNIX-Systemen).

Für Dateien im Format Variabel geben Sie die maximale Satzlänge an. Siehe Beispiel für Parameter Format.

-I <input code> Für ASCII(ANSI) - EBCDIC Umwandlung stehen folgende Konvertierungstabellen zur Verfügung: ASCII-IBM037, ASCII-IBM237, ANSI-IBM037, ANSI-IBM273.

Für EBCDIC - ASCII(ANSI) Umwandlung stehen folgende Konvertierungstabellen zur Verfügung:
IBM037-ASCII, IBM237-ASCII, IBM037-ANSI, IBM273-ANSI.

Beispiel: Wenn Sie die Umwandlungstabelle IBM237-ASCII benutzen möchten, geben Sie als Input Code IBM237 (`-I IBM237`) und als Output Code ASCII (`-O ASCII`) an.

-O <output code> siehe Input Code.

-t <conversion table> Pfad und der Name Ihrer eigenen Codeumwandlungstabelle. Lesen Sie bitte das rvsXP Benutzerhandbuch, Kapitel „Codeumwandlung“ für mehr Informationen.

-S <serialize>	**Y**(Yes)/**N**(No) Diese Option gewährleistet, dass Ihre Dateien in einer geordneten Reihenfolge ankommen. Alle Dateien, die in der gleichen Gruppe versendet werden sollen, müssen die gleiche Kennung (Label) haben. Siehe `-l <label>`.
-l <label>	Alle Dateien, die in der gleichen Gruppe versendet werden, müssen die gleiche Kennung (Label) haben. Siehe auch Option `-s <serialize>`.
-za <attempts>	Anzahl von Sendeversuchen für synchronisierte Übertragung.
-zt <timeout>	Timeout in Sekunden für synchronisierte Übertragung.

Optionale EDI-Konvertierungsparameter:

-cf <Dateipfad<	Format-Beschreibung für den ersten Schritt des EDI-Konverters
-cf2 <Dateipfad<	Format-Beschreibung für den zweiten Schritt des EDI-Konverters (optional)
-ct <Dateipfad>	Stylesheet für den ersten Schritt des EDI-Konverters
-cd	EDI-Konverter-Richtung für den ersten Schritt, default EDI2XML
-cd2	EDI-Konverter-Richtung für den zweiten Schritt, default XML2EDI.
-cl <Ganzzahl>	log level für den EDI-Konverter, default:0
-cs <0|1>	Verhalten am Zeilenende, für die EDI-Nachricht-Ausgabe 0 - kein LF als Segment-Trenner 1 - mit LF als Segment-Trenner
-cl <0|1>	Einrückung für XML-Ausgabe; 0 - ohne Einrückung; 1 - mit Einrückung (default).
-ce <encoding>	Encoding für die XML-Ausgabe; default UTF-8.

Optionale Parameter:

-verbose	Ausgabe von ausführlichen Meldungen.

© gedas deutschland GmbH / Fachbereich rvs® Systems

-help	Gibt Beschreibung zum aktuellen Kommando aus
-?	Fordert Hilfe an.

Beispiele:

```
convertAndSend -d C:\teil56.txt -s RVS -v TEILE
-za 4
```

In diesem Beispiel wird die Datei `C:\teil56.txt` zur rvs® - Zentrale mit dem virtuellen Dateinamen `TEILE` gesendet; die Anzahl der Sendeversuche ist begrenzt auf 4.

```
convertAndSend
-d C:\INTEGRATION\test.txt -s RVS -v TEST
-cf C:\rvsET\system\fmtDesc\fw.kanban.ineas.xml
-ct C:\rvsET\system\stylesheets\ineas2deljit.xslt
-cf2 C:\rvsET\system\fmtDesc\edifact.97.orig.xml
```

In diesem Beispiel wird die Datei `C:\INTEGRATION\test.txt` zur rvs® - Zentrale mit dem virtuellen Dateinamen TEST gesendet. Die Datei wurde auch mit dem EDI-Konverter **WEDIConv** (der in dem Verzeichnis C:\rvsET installiert wurde) konvertiert, zuerst vom Inhouse-Format (Datei `test.txt`) zum XML-Format (Datei `fw.kanban.ineas.xml`) und dann zur EDIFACT - Nachricht (Datei `edifact.97.orig.xml`). Beim ersten Schritt wurde für die Darstellung folgendes Stylesheet benutzt: `ineas2deljit.xslt`.

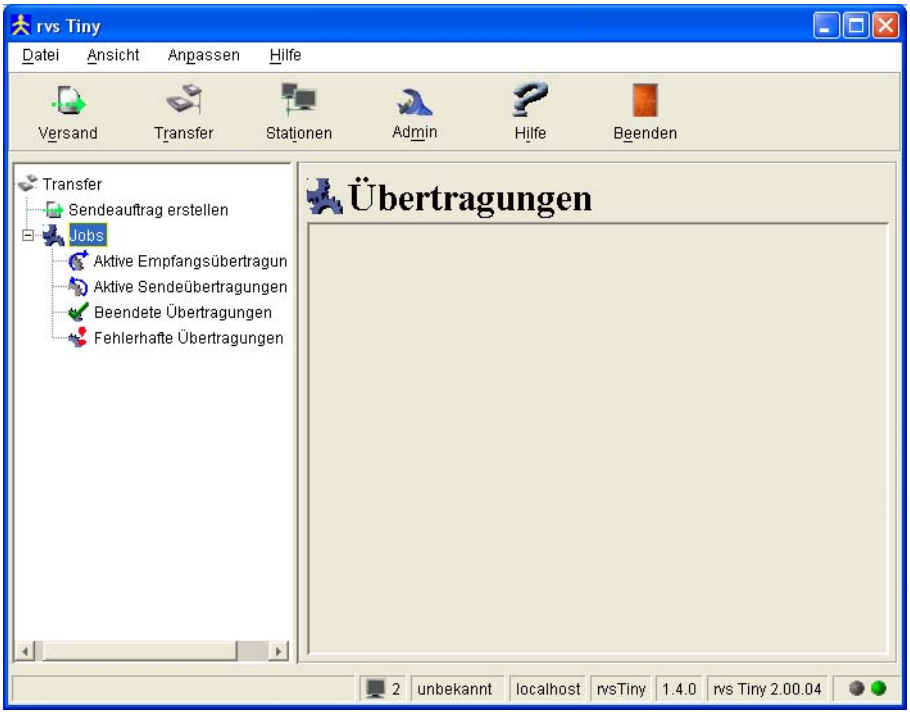
Hinweis: Bei der Benutzung dieses Programms ist zu beachten, dass ein erneuter Sendeauftrag nach Fehler doppelte Übertragungen zur Folge haben kann. Wenn z.B. die Datei schon zur direkten Gegenstelle übertragen werden konnte, jedoch innerhalb der vorgegebenen Übertragungszeit die Quittung nicht eintraf, ist die Übertragung unabhängig von der eigenen Versandstation weiterhin aktiv. Es kann sein, dass die Datei von unserer direkten Partnerstation trotzdem noch übertragen wird. Darauf hat die eigene Station keinen Einfluss mehr. Ein erneutes Versenden der Datei hat dann die doppelte Übertagung zur Folge. Die Anwendung in der Schicht über rvs® muss also mit einer solchen Situation umgehen können. In der Regel wird dies gelöst, indem die Datei anhand des Dateinamens eindeutig ist, z.B. durch Verwendung eines aufsteigenden Zählers oder indem die Anwendung die Eindeutigkeit über den Dateiinhalt feststellt.

4.7 Ausgabe aller Empfangs- und Sendejobs

Die wichtigsten Informationen über die Sende- und Empfangsjobs können über die GUI oder mit dem Program `getJobList` angezeigt werden. Wie Sie die rvs® Tiny GUI starten können, lesen Sie bitte im Kapitel 2.5.

GUI

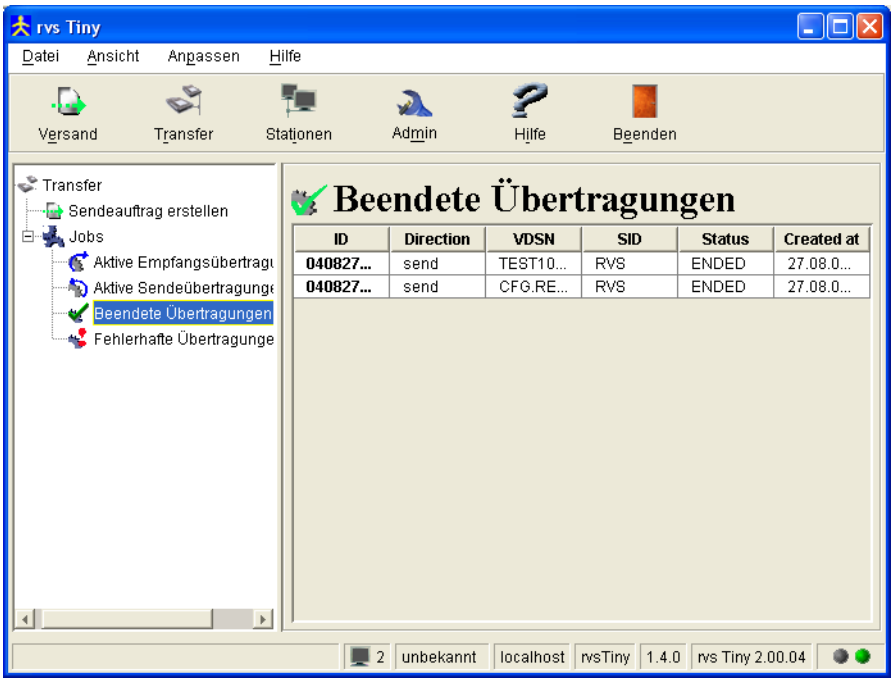
Wenn Sie das Symbol Transfer in der Funktionsleiste wählen, öffnet sich das Fenster mit dem Ordner Jobs in dem Transferbaum. Sie können jetzt zwischen Aktiven Empfangs/Sendeübertragungen, Beendeten Übertragungen (Sende- und Empfangs-Richtung) und Fehlerhaften Übertragungen wählen.



Hinweis: Eine Übertragung gilt erst als abgeschlossen, wenn die ODETTE-Quittung EERP (End-to-End-Response) für diese Übertragung angekommen ist.

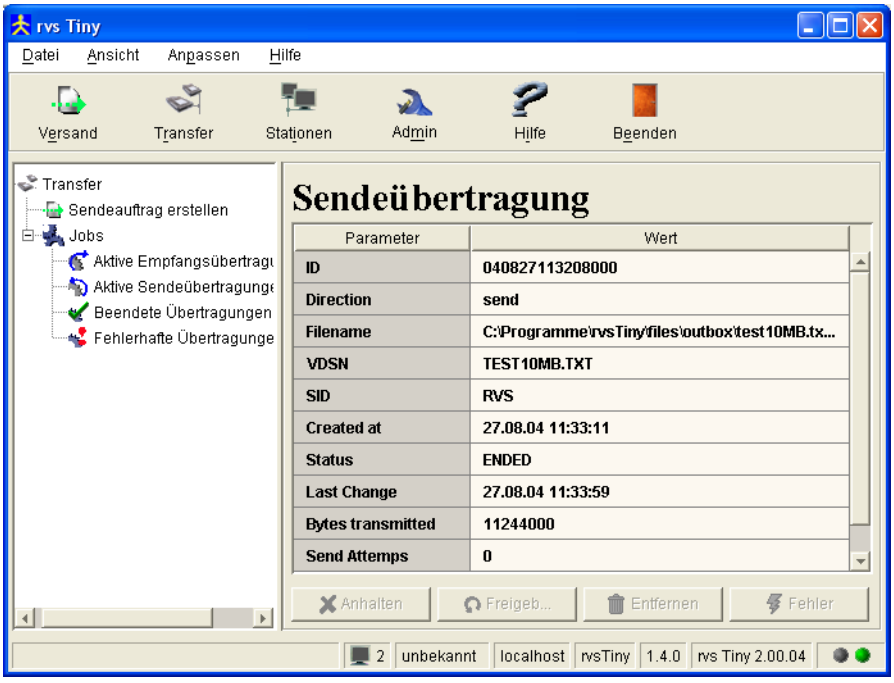
Eine Übersicht aller vorhandenen Jobs bekommen Sie, indem Sie den gewünschten Job-Ordner mit einem Klick anwählen (markieren).

© gedas deutschland GmbH / Fachbereich rvs® Systems



Die Daten eines Jobs können Sie sich anzeigen lassen, in dem Sie mit der Maus einen Doppelklick auf eine Jobzeile im rechten Teil des Fensters ausführen.

Beispiel



Bei den aktiven Empfangsübertragungen bietet rvs® Tiny die Möglichkeit den EERP anzuhalten, freizugeben oder zu löschen. Einen EERP anhalten können Sie, wenn Sie im rechten Fensterteil die Schaltfläche **Anhalten** betätigen; wieder freigeben mit der Schaltfläche **Freigeben** und löschen mit der Schaltfläche **Entfernen**. Je nachdem für welche Aktion Sie sich entscheiden, ändert sich der Wert in der Zeile Status (z.B. EERP_HOLDED). Das Gleiche können Sie auch mit dem Programm `handleEERP` erreichen (siehe Kapitel 4.9).

Bei den aktiven Sendeübertragungen haben die gleichen Schaltflächen im unteren Teil des einzelnen Jobfensters (Detailansicht) die Aufgabe, einen Sendejob anzuhalten, zu löschen oder wieder freizugeben. Ein aktiver Sendejob muss zuerst angehalten werden, um gelöscht werden zu können.

Eingabeaufforderung

`getJobList` Mit dem Programm `getJobList` können Sie alle Jobs auflisten.

Syntax:

```
getJobList [-a][-e][-f][-verbose]
```

Optionale Parameter:

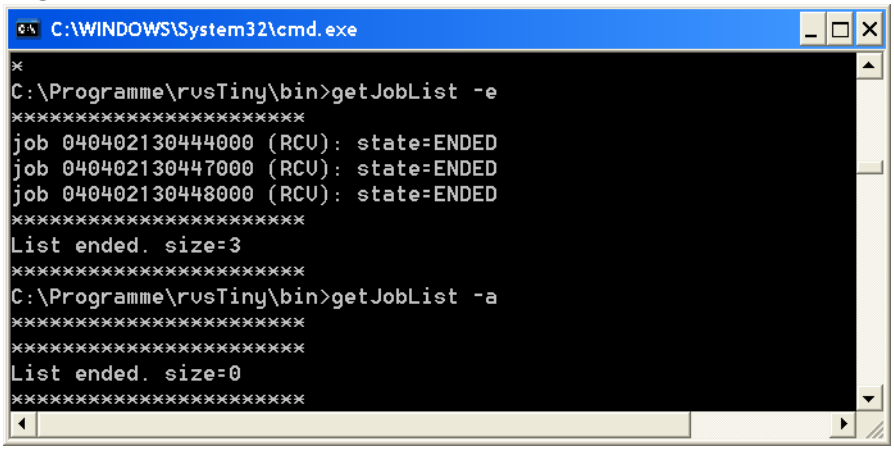
-a	Ausführliche Ausgabe der Informationen über Jobs, die in Bearbeitung sind.
-e	Ausgabe der Informationen über beendete Jobs.
-f	Ausgabe der Informationen über fehlgeschlagene Jobs.
-h	Fordert Hilfe (Usage) an.
-verbose	Ausgabe von ausführlichen Meldungen.
-?	Fordert Hilfe (Usage) an.

Beispiel:

```
getJobList -e
```


© gedas deutschland GmbH / Fachbereich rvs® Systems

Ergebnis:


C:\WINDOWS\System32\cmd.exe
x
C:\Programme\rvsTiny\bin>getJobList -e
xxxxxxxxxxxxxxxxxxxxxxx
job 040402130444000 (RCV): state=ENDED
job 040402130447000 (RCV): state=ENDED
job 040402130448000 (RCV): state=ENDED
xxxxxxxxxxxxxxxxxxxxxxx
List ended. size=3
xxxxxxxxxxxxxxxxxxxxxxx
C:\Programme\rvsTiny\bin>getJobList -a
xxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxx
List ended. size=0
xxxxxxxxxxxxxxxxxxxxxxx

4.8 Ausgabe eines JobEintrages

`getJob` Mit dem Programm `getJob` erhalten Sie Informationen zu einem bestimmten Sende- oder Empfangsjob.

Syntax:

```
getJob -n <jobid> [-a] [-verbose]
```

Benötigte Parameter:

-n <jobid>	Angabe eines Sende- oder Empfangsjobs mit der ID `<jobid>`.

Optionale Parameter:

-a	Ausgabe der komplett verfügbaren Job-Informationen.
-help	Fordert Hilfe (Usage) an.
-verbose	Ausgabe von ausführlichen Meldungen.
-?	Fordert Hilfe (Usage) an.

Beispiel:

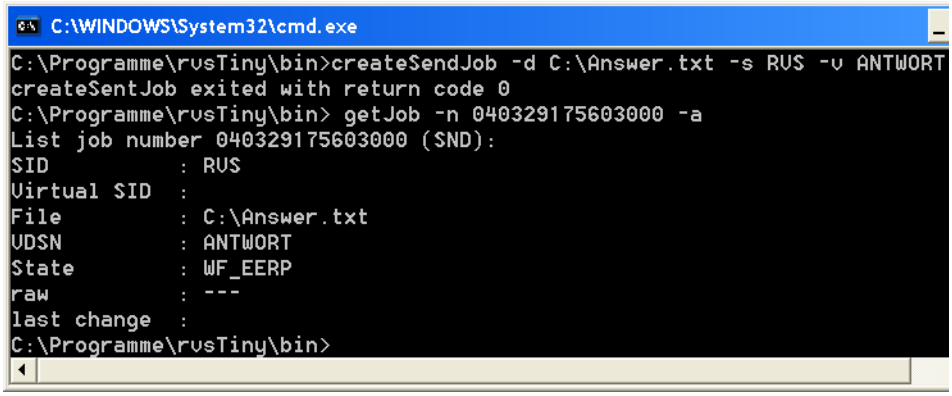
```
getJob -n 040329175603000
```

Ergebnis: `job 040329175603000 (SND): state: WF_EERP`

Beispiel:

```
getJob -n 040329175603000
```

Ergebnis:



© gedas deutschland GmbH / Fachbereich rvs® Systems

4.9 Löschen oder Freigeben von EERPs

`handleEERP` Mit dem Programm `handleEERP` können Sie Empfangbestätigungen (EERP_OUT) löschen oder freigeben.

Syntax:

```
handleEERP -r|-d <num> [-verbose]
```

Benötigte Parameter:

-r | -d <num>	ID des Jobs für welchen die EERP freigegeben / gelöscht werden soll.

Optionale Parameter:

-help	Fordert Hilfe (Usage) an.
-verbose	Ausgabe von ausführlichen Meldungen.
-?	Fordert Hilfe (Usage) an.

4.10 Archivieren der Einträge der abgearbeiteten Sende- bzw. Empfangsaufträge im Revisionslog

`archiveJobs` Mit dem Programm `archiveJobs` wird im Verzeichnis `$RVSTINY_HOME/rvsTiny/archive` eine Datei `revision.log` erzeugt, in der Einträge der erfolgreich abgearbeiteten Sende- bzw. Empfangsaufträge gespeichert sind.

Syntax:

```
archiveJobs [-verbose] [-help] [-?]
```

Optionale Parameter:

-help	gibt Beschreibung zum aktuellen Kommando aus.
-verbose	Ausgabe von ausführlichen Meldungen.
-?	Fordert Hilfe (Usage) an.

Dabei werden beendete Jobs aus dem Verzeichnis `$RVSTINY_HOME\rvsTiny\jobs\ENDED` gelöscht und die Daten in die Datei `revision.log` im Verzeichnis `$RVSTINY_HOME\rvsTiny\archive` geschrieben.

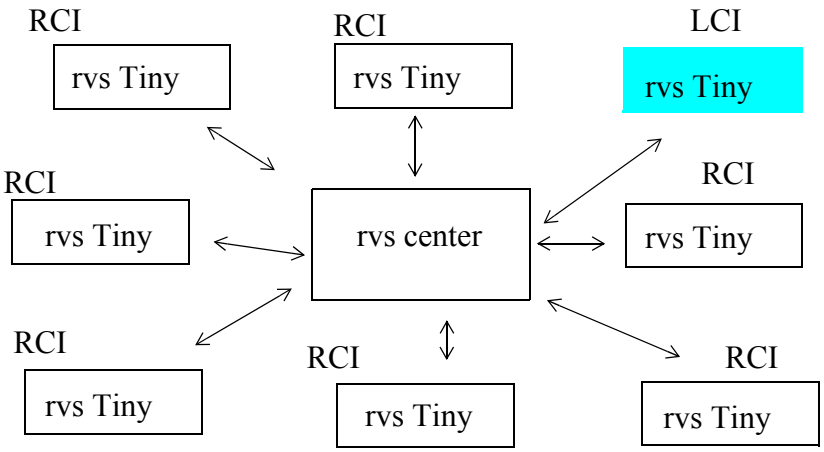
5 Zentrale Administration von rvs® Tiny

In diesem Kapitel ist eine mächtige Funktionalität von rvs® Tiny beschrieben: wie man mit einer rvs® Tiny-Installation andere rvs® Tiny Stationen administrieren kann. rvs® Tiny ist die ideale Software für eine Sterntopologie, in der viele Stationen mit einem zentralen Server (rvs® Zentrale) verbunden sind. Aus diesem Grunde ist die Zentrale Administration von besonderer Bedeutung für den rvs® Tiny-Netzwerk-Administrator, da sie ihm ermöglicht, alle rvs® Tiny-Installationen entfernt zu pflegen.

5.1 Einleitung

Die Zentrale Administration von rvs® Tiny ermöglicht das Administrieren von zahlreichen rvs® Tiny-Installationen mit Hilfe spezieller Konfigurationsdateien.

Die Konfigurationsdateien werden von der rvs® Tiny-Station gesendet, die andere rvs® Tiny - Stationen administriert (sie wird in diesem Buch Local Configuration Instance, LCI, genannt) zu der Station, die administriert weden soll (wir nennen sie: Remote Controlled Instance RCI). Die beiden Stationen (Instanzen) befinden sich in dem rvs® Tiny-Stern - Netzwerk (Siehe die folgende Abbildung).



Die Konfigurationsdateien für die Zentrale Administration nennen wir **Container-Konfigurationsdateien**, um sie von den normalen rvs® Tiny XML-Konfigurationsdateien wie z.B. `rvsTinyStationlist.xml` zu unterscheiden.

Container-Konfigurationsdateien sind:

- `cfg.req.jar` für Konfigurationsanfragen, die LCI an RCI sendet
- `cfg.rsp.jar` für Konfigurationsantworten, die RCI an LCI sendet.

© gedas deutschland GmbH / Fachbereich rvs® Systems

Die Local Configuration Instance LCI

Jede rvs® Tiny-Installation kann eine LCI werden, wenn Sie während der Installation von rvs® Tiny im Fenster „Choose Product Feature“ die „Complete Installation“ wählen (siehe Kapitel 2.3).

Die rvs® Tiny-Installation, die zur LCI geworden ist, besitzt dann ein spezielles Verzeichnis, wo Konfigurationsdaten von allen RCIs (rvs® Tiny - Stationen, die entfernt administriert werden sollen) gespeichert werden - das ist das Configuration Repository CRep. Das CRep-Verzeichnis ist `$RVSTINY_HOME/management`. Siehe bitte Kapitel 1.2 für eine detaillierte Erklärung von `$RVSTINY_HOME`.

Das CRep (`$RVSTINY_HOME/management`) besteht aus folgenden Verzeichnissen oder Dateien:

– `$RVSTINY_HOME/management/mgmt-datastore`
  Dieses Verzeichnis entsteht nur nach einer erfolgreichen Übertragung und erhaltenen Antwort auf eine Anforderung (request). Die Anforderung einer Container-Konfigurationsdatei stellt eine LCI an eine RCI. Siehe bitte Kapitel 5.3 für die Erklärung, wie eine Anforderung erzeugt wird.

– `$RVSTINY_HOME/management/mgmt-log/activity.log`
  Diese log-Datei enthält Protokolle allller Konfigurationsschritte. Jeder Eintrag in dieser Datei besteht aus einem Zeitstempel, dem Typ der Konfiguration: **configuration request** (Konfigurations-Anforderung) oder **configuration response** (Konfigurations-Antwort), der SID der administrierten rvs® Tiny-Station (RCI) und der Aktivitätsmeldung.

– `$RVSTINY_HOME/management/mgmt-templates`
  Dieses Verzeichnis enthält templates für die Aktionen, die der rvs® Tiny-Netzwerk-Administrator durchführt. Sie brauchen dieses Verzeichnis nur wenn Sie die Administrator-Software updaten.

– `$RVSTINY_HOME/management/mgmt-workspace`
  In diesem Verzeichnis sind Konfigurationsdateien von RCIs, die administriert werden sollen, gespeichert. Das Verzeichnis enthält Unterverzeichnisse, die nach der RCIs SID bennant werden (Siehe bitte das Programm `prepareUpdateStation`, Kapitel 5.3 als Beispiel).

Die Remote Controlled Instance RCI

Eine RCI ist die rvs® Tiny-Station, die entfernt administriert werden soll. Als Installationstyp für eine RCI empfiehlt sich „Typical Installation“ (Siehe Kapitel 2.3) und nicht „Complete Installation“ wie für eine LCI.

Wenn eine rvs® Tiny-RCI eine Konfigurationsanforderung (Datei `cfg.req.jar`) von einer LCI empfängt, löst dieser Vorgang den Start eines entsprechenden Jobs aus. Dieser Job verarbeitet die Konfigurationsanforderung und generiert eine Konfigurationsantwort. Die Konfigurationsantwort wird dann an die LCI als Datei `cfg.rsp.jar` zurückgesendet.

5.2 Programme der Zentralen Administration

Der ganze Prozess (alle Konfigurations- und Administrationsfälle) sind mit den folgenden Programmen auszuführen. Diese Programme befinden sich im Verzeichnis `$RVSTINY_HOME\bin` als Batch-Dateien (Siehe Kapitel 1.2 für eine detaillierte Erklärung von `$RVSTINY_HOME`.). Um sie ausführen zu können, müssen Sie sie aus dem Verzeichnis `$RVSTINY_HOME\bin` starten.

orderConfiguration.bat -s <SID>	holt die Konfiguration von einer RCI und speichert diese ins CRep-Verzeichnis. **Beispiel** (für CRep): `C:\rvsTiny\management\mgmt-datastore\TINYPW` In diesem Beispiel ist `TINYPW` die SID von der RCI (von der Station, die administriert werden soll).
prepareUpdateStation.bat -s <SID>	holt eine Kopie der RCI-Konfiguration aus dem CRep-Verzeichnis ins WorkDir, um dort die Konfiguration durchzuführen. **Beispiel** für WorkDir: Im Verzeichnis `C:\rvsTiny \management\mgmt-workspace` wird folendes Unterverzeichnis erzeugt: `TINYPW\UPDATE_STATION_040826_114418\out`. Das ist das Unterverzeichnis für die Konfiguration der Station `TINYPW`; Datum der Erzeugung ist 2004-08-26, Zeit 11:44:18.
commitUpdateStation.bat -s <SID> -d <WorkDir>	sendet die modifizierte Konfiguration aus dem WorkDir (ohne Unterverzeichnis `out`) zur RCI (Die RCI soll in der Option `-s` für die StationsID angegeben werden). **Beispiel**: `commitUpdateStation -s TINYPW -d C:\rvsTiny\management\mgmt-workspace\TINYPW\UPDATE_STATION_040826_114418`

Hinweis: Der Name des Verzeichnisses WorkDir besteht aus dem CRep (Siehe Kapitel 5.1); der SID der RCI (`TINYPW` im obigen Beispiel), dem Verzeichnis `UPDATE_STATION` mit dem Zeitstempel (der wiederum aus Datum und Uhrzeit besteht; im Beispiel `040826_114418`) und dem Verzeichnis `out`.

© gedas deutschland GmbH / Fachbereich rvs® Systems

WorkDir enthält alle wichtigsten rvs® Tiny-Verzeichnisse und Dateien: Hier ist ein Beispiel, wie Sie die Konfigurationsdateien ändern können:

– Ändern Sie die Stationskonfiguration, indem Sie die Datei `$RVSTINY_HOME/conf/rvsTinyStationlist.xml` editieren.

– Ändern Sie die Jobstarts-Konfiguration, indem Sie die Datei `$RVSTINY_HOME/conf/rvsTinyJobstart.xml` editieren.

– Machen Sie ein Update der Software, indem Sie die .jar-Dateien im Verzeichnis `$RVSTINY_HOME/lib` austauschen.

5.3 Wie arbeite ich mit der Zentralen Administration?

Die folgenden Schritte sollten Ihnen als typisches Konfigurationsbeispiel dienen. Alle diese Schritte sind notwendig, unabhängig davon, ob Sie ein Update durchführen, einen Lizenzschlüssel austauschen oder die Konfiguration der Stationen, Parameter oder Jobstarts ändern möchten. Mehr Details über derartige Administrationsaufgaben erhalten Sie in den Kapiteln 5.3.1, 5.3.2 und 5.3.3.

Hinweis: Die folgenden Befehle sind als Batch-Dateien im Verzeichnis `$RVSTINY_HOME\bin` vorhanden (Siehe bitte Kapitel 1.2 für eine detaillierte Erklärung von `$RVSTINY_HOME`.), so dass Sie sich in diesem Verzeichnis befinden müssen, um diese Programme ausführen zu können.

- Der erste Schritt ist immer, die Konfiguration der zu administrierenden Station, RCI, anzufordern. Für diesen Zweck dient das Programm `orderConfiguration`.

  Beispiel: Wenn Sie die Konfiguration der rvs® Tiny Station `TINY11` ändern möchten, müssen Sie in der Eingabeaufforderung (Kommandozeile) das Programm `orderConfiguration` mit dem folgenden Befehl starten:

  ```
  orderConfiguration -s TINY11
  ```

  Dieser Befehl erzeugt die Konfigurationsanforderungsdatei `cfg.req.jar` (Siehe Kapitel 5.1 für die Erklärung der `cfg.req.jar`) und sendet diese Datei via OFTP zur Station `TINY11`. Die Übertragung der Datei `cfg.req.jar` können Sie in der rvs® Tiny GUI (Admin-Fenster) verfolgen. Mit dem Empfang der Datei `cfg.req.jar` bei der RCI (`TINY11`), wird ein Konfigurationsprozess angestoßen. Dieser Prozess stoppt rvs® Tiny (bei der Station `TINY11`), archiviert die aktuelle Konfiguration in der Datei `cfg.rsp.jar`, startet rvs® Tiny (`TINY11`) wieder und sendet die Konfigurationsantwortdatei `cfg.rsp.jar` zurück zur LCI (rvs® Tiny-Station des Administrators). Im Falle, dass dieser Schritt erfolgreich war, bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“. Alle diese

Schritte führt das Programm `orderConfiguration` automatisch aus.

Der nächste Schritt ist das Kopieren der RCI-Konfiguration, die als Datei `cfg.rsp.jar` empfangen wurde. Das ist die Aufgabe des Programms `prepareUpdateStation`. Dieses Programm kopiert für Sie die empfangenen Konfigurationsantwortdatei `cfg.rsp.jar` und speichert diese ins WorkDir-Verzeichnis (Siehe Kapitel 5.2 für die Erklärung von WorkDir).

Beispiel:

```
prepareUpdateStation -s TINY11
```

Ergebnis: Das Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY11\UPDATE_STATION_040828_113315\out` mit der kompleten Konfiguration der Station `TINY11` wird erzeugt. Die Meldung über den Erfolg oder Misserfolg dieses Schrittes finden Sie in der Datei `activity.log`. Diese log-Datei befindet sich im Verzeichnis `$RVSTINY_HOME/management/mgmt-log`.

- Jetzt können Sie die Konfiguration der RCI (rvs® Tiny-Station `TINY11` im Beispiel) ändern. Sie können z.B. einen Lizenzschlüssel austauschen, die XML-Konfigurationsdatei für Stationen oder Jobstarts editieren oder durch das Ersetzen der ensprechenden .jar-Datei ein Update von rvs® Tiny durchführen. Mehr Details über derartige Administrationsaufgaben erhalten Sie in den Kapiteln 5.3.1, 5.3.2 und 5.3.3.
- Senden Sie die modifizierte Konfiguration zur RCI (`TINY11`). Sie müssen das ganze Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY11\UPDATE_STATION_040828_113315)` mit dem folgenden Befehl senden:

  ```
  commitUpdateStation -s TINY11 -d
  C:\rvsTiny\management\mgmt-
  workspace\TINY11\UPDATE_STATION_040828_113315
  ```

  Dieser Befehl wird das ganze geänderte Vereichnis zusammenfassen und zur RCI (`TINY11`) zurücksenden unter dem Dateinamen `cfg.req.jar`.

  Nachdem die RCI-Station (`TINY11`) die Datei `cfg.req.jar` erhalten hat, wird rvs® Tiny gestoppt (alle diese Schritte werden mit durch das Programm `commitUpdateStation` ausgeführt, Sie brauchen keine zusätzlichen Aktionen zu starten) und die geänderte Konfiguration wird aktualisiert. Der Job prüft zusätzlich, ob alles fehlerfrei durchgeführt wurde und sendet die Datei `cfg.rsp.jar` als Antwort zurück. Das Ergebnis dieser Aktion wird auch in der Datei `activity.log` protokolliert.

Hinweis: Im Falle eines Misserfolgs wird die alte Konfiguration wieder aktiviert.

© gedas deutschland GmbH / Fachbereich rvs® Systems

5.3.1 Wie tausche ich einen Lizenzschlüssel aus?

Dieses Kapitel beschreibt einen typischen Fall bei der Administration von rvs® Tiny-Stationen, nämlich wie man einen nicht mehr gültigen Lizenzschlüssel austauscht. Im nächsten Beispiel wird die Station `TINY01` die Station `TINY02` administrieren.

Voraussetzungen: Die Station `TINY01` muss die Station `TINY02` in der Stationstabelle als eine geroutete Station via die rvs®-Zentrale haben (Siehe bitte Kapitel 3.2.1 für die Erklärung, wie man Stationen anlegt) und die Station `TINY02` muss die Station `TINY01` als eine geroutete Station via die rvs® - Zentrale haben. Die Station `TINY01` muss eine „Complete Installation“ sein und für die Station `TINY02` ist „Typical Installation“ ausreichend (Siehe bitte Kapitel 2.3 für die Erklärung der „Typical“ und „Complete“ Installation).

- Der erste Schritt, um einen Lizenzschlüssel auszutauschen, ist mit dem folgenden Befehl in der Eingabeaufforderung die Konfiguration von `TINY02` anzufordern:

  ```
  orderConfiguration -s TINY02
  ```

  Bei einem Erfolg dieser Aktion, bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“ und anschließend werden Sie als Antwort auf Ihre Konfigurationsanforderung die Datei `cfg.rsp.jar` erhalten. (Das können Sie bei den **Beendeten Übertragungen** im Admin-Fenster von der `TINY01` GUI verfolgen.

- Der nächste Schritt ist, eine Kopie der `TINY02`-Konfiguration zu erzeugen (diese Konfiguration ist in der Datei `cfg.rsp.jar` enthalten). Dazu wird der folgende Befehl benutzt:

  ```
  prepareUpdateStation -s TINY02
  ```

  Ergebnis: Das Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315\out` mit der vollständigen Konfiguration der Station `TINY02` wurde erzeugt. Das Ergebnis dieser Aktion wird in der Datei `activity.log` protokolliert. Diese log-Datei befindet sich im Verzeichnis `$RVSTINY_HOME/management/mgmt-log`.

- Jetzt können Sie den alten Lizenzschlüssel aus dem Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315\out\conf` nach `licenseOLD.properties` umbenennen und den neuen Lizenzschlüssel `license.properties` ins Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315\out\conf` kopieren. Wie Sie einen neuen Lizenzschlüssel erwerben können, lesen Sie bitte im Kapitel 2.2.

- Senden Sie die geänderte Konfiguration zur Station TINY02 (Sie müssen das ganze Verzeichnis C:\rvsTiny\management\mgmt-workspace\TINY02\UPDATE_STATION_040828_113315) mit dem folgenden Befehl senden:

  ```
  commitUpdateStation -s TINY02 -d
  C:\rvsTiny\management\mgmt-
  workspace\TINY02\UPDATE_STATION_040828_113315
  ```

  Dieser Befehl wird das ganze geänderte Verzeichnis zusammenfassen und unter dem Dateinnamen cfg.req.jar an die Station TINY02 senden.

- Nach erfolgreichem Empfang der Datei cfg.req.jar bei der Station TINY02, wird rvs® Tiny von TINY02 gestoppt (alle diese Schritte werden vom Programm commitUpdateStation selbständig ausgeführt); die geänderte Konfiguration wird aktualisiert; der Job prüft noch, ob alles fehlerfrei abgelaufen war und sendet die Antwort unter dem Dateinamen cfg.rsp.jar zurück. Das Ergebnis dieser Änderungen wird in der Datei activity.log protokolliert.

Hinweis: Im Falle eines Misserfolgs wird auf der RCI die alte Konfiguration wieder eingespielt.

5.3.2 Wie ändere ich die Konfiguration einer Station?

Dieses Kapitel beschreibt einen typischen Fall in der Administration von rvs® Tiny, nämlich wie man Stationsparameter z.B. die ODETTE-ID ändern kann. Die gleiche Vorgehensweise gilt auch für andere Parameter (andere Stationsparameter, Jobstartparameter oder rvs® Tiny globale Parameter). In diesem Beispiel wird die Station TINY20 die Station TINY22 administrieren.

Voraussetzungen: Die Station TINY20 muss die Station TINY22 in der Stationstabelle als eine geroutete Station via die rvs®-Zentrale haben (Siehe bitte Kapitel 3.2.1 für die Erklärung, wie man Stationen anlegt) und die Station TINY22 muss die Station TINY20 als eine geroutete Station via die rvs® - Zentrale haben. Die Station TINY20 muss eine „Complete Installation“ sein und für die Station TINY22 ist „Typical Installation“ ausreichend (Siehe bitte Kapitel 2.3 für die Erklärung der „Typical“ und „Complete“ Installation).

- Um die Station von TINY22 entfernt konfigurieren zu können, muss die Station TINY20 zuerst die Konfiguration von TINY22 anfordern. Das geschieht mit dem folgenden Befehl:

  ```
  orderConfiguration -s TINY22
  ```

  Bei einem Erfolg dieser Aktion bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“ und anschließend werden Sie als Antwort auf Ihre Konfigurationsanforderung die Datei cfg.rsp.jar erhalten. (Das

© gedas deutschland GmbH / Fachbereich rvs® Systems

können Sie bei den **Beendeten Übertragungen** im Admin-Fenster von der TINY20 GUI verfolgen.

- Der nächste Schritt ist, eine Kopie der TINY22-Konfiguration im WorkDir-Verzeichnis von TINY20 zu erzeugen (diese Konfiguration ist in der Datei cfg.rsp.jar enthalten). Dazu wird der folgende Befehl benutzt:

  ```
  prepareUpdateStation -s TINY22
  ```

  Ergebnis: Das Verzeichnis C:\rvsTiny\management\mgmt-workspace\TINY22\UPDATE_STATION_040828_113315\out mit der vollständigen Konfiguration der Station TINY22 wurde erzeugt. Das Ergebnis dieser Aktion wird in der Datei activity.log protokolliert. Diese log-Datei befindet sich im Verzeichnis $RVSTINY_HOME/management/mgmt-log.

- Jetzt können Sie die Datei rvsTinyStationlist.xml aus dem Verzeichnis C:\rvsTiny\management\mgmt-workspace\TINY22\UPDATE_STATION_040829_133315\out\conf editieren und z.B. den Parameter ODETTE_ID oder TCP/IP-Adresse (Parameter IP_ADDR) für die lokale Station von TINY22 (STATION_LOC) oder für andere Stationen (rvs® - Zentrale oder geroutete Stationen) ändern.

- Der nächste Schritt ist, die geänderte TINY22-Konfiguration an diese Station zu senden. Sie müssen das ganze Verzeichnis (aber ohne Unterverzeichnis out) C:\rvsTiny\management\mgmt-workspace\TINY02\UPDATE_STATION_040829_133315) mit dem folgenden Befehl senden:

  ```
  commitUpdateStation -s TINY22 -d
  C:\rvsTiny\management\mgmt-
  workspace\TINY22\UPDATE_STATION_040829_133315
  ```

  Dieser Befehl wird das ganze modifizierte Verzeichnis zusammenfassen und unter dem Dateinamen cfg.req.jar an die Station TINY22 senden.

- Nach dem erfolgreichen Empfang der Datei cfg.req.jar bei der Station TINY22, wird rvs® Tiny auf TINY22 gestoppt (alle diese Schritte werden vom Programm commitUpdateStation selbständig ausgeführt); die geänderte Konfiguration wird aktualisiert; der Job prüft noch, ob alles fehlerfrei abgelaufen war und sendet die Antwort unter dem Dateinamen cfg.rsp.jar zurück. Das Ergebnis dieser Änderungen wird auf der LCI in der Datei activity.log protokolliert.

Hinweis: Im Falle eines Misserfolgs wird auf der RCI die alte Konfiguration wieder eingespielt.

5.3.3 Wie führe ich ein Update von rvs® Tiny durch?

Dieses Kapitel beschreibt einen typischen Fall in der Administration von rvs® Tiny-Stationen, nämlich wie man ein Update einer anderen rvs® Tiny - Station durchführt. In diesem Beispiel wird die Station `TINY30` die Station `TINY33` administrieren.

Voraussetzungen: Die Station `TINY30` muss die Station `TINY33` in der Stationstabelle als eine geroutete Station via die rvs®- Zentrale haben (Siehe bitte Kapitel 3.2.1 für die Erklärung, wie man Stationen anlegt) und die Station `TINY33` muss die Station `TINY30` als eine geroutete Station via die rvs® - Zentrale haben. Die Station `TINY30` muss eine „Complete Installation“ sein und für die Station `TINY33` ist „Typical Installation“ ausreichend (Siehe bitte Kapitel 2.3 für die Erklärung der „Typical“ und „Complete“ Installation).

- Um die Station `TINY33` updaten zu können, muss die Station `TINY30` zuerst die Konfiguration von `TINY33` anfordern. Das geschieht mit dem folgenden Befehl:

  ```
  orderConfiguration -s TINY33
  ```

  Bei einem Erfolg dieser Aktion bekommen Sie in der Eingabeaufforderung die Meldung „OrderConfiguration exited with return code *0*“ und anschließend werden Sie als Antwort auf Ihre Konfigurationsanforderung die Datei `cfg.rsp.jar` erhalten. (Das können Sie bei den **Beendeten Übertragungen** im Admin-Fenster von der `TINY30` GUI verfolgen.

- Der nächste Schritt ist, eine Kopie der `TINY33`-Konfiguration im WorkDir-Verzeichnis von `TINY30` zu erzeugen (diese Konfiguration ist in der Datei `cfg.rsp.jar` enthalten). Dazu wird der folgende Befehl benutzt:

  ```
  prepareUpdateStation -s TINY33
  ```

  Ergebnis: Das Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY33\UPDATE_STATION_040830_113315\out` mit der vollständigen Konfiguration der Station `TINY33` wurde erzeugt. Das Ergebnis dieser Aktion wird in der Datei `activity.log` protokolliert. Diese log-Datei befindet sich im Verzeichnis `$RVSTINY_HOME/management/mgmt-log`.

- Benennen Sie jetzt die alte .jar-Datei `rvstiny.jar` aus dem Verzeichnis `C:\rvsTiny\management\mgmt-workspace\TINY33\UPDATE_STATION_040830_113315\out\lib` nach `rvstinyOLD.jar` um und ersetzen Sie die alte .jar-Datei durch eine neue. Um eine aktuelle .jar-Dateie für das Update zu bekommen, wenden Sie sich bitte an den rvs® - Kundendienst (E-Mail: rvs-service@gedas.de; Tel. +49 30 39971 777; fax: +49 30 39971 994).

- Senden Sie ist die geänderte `TINY33`-Konfiguration zu dieser Station. Sie müssen das ganze Verzeichnis (aber ohne

© gedas deutschland GmbH / Fachbereich rvs® Systems

Unterverzeichnis `out`) `C:\rvsTiny\management\mgmt-workspace\TINY33\UPDATE_STATION_040829_133315`) mit dem folgenden Befehl senden:

```
commitUpdateStation -s TINY33 -d
C:\rvsTiny\management\mgmt-
workspace\TINY02\UPDATE_STATION_040830_113315
```

Dieser Befehl wird das ganze modifizierte Verzeichnis zusammenfassen und unter dem Dateinamen `cfg.req.jar` an die Station `TINY33` senden.

- Nach dem erfolgreichen Empfang der Datei `cfg.req.jar` bei der Station `TINY33`, wird rvs® Tiny auf `TINY33` gestoppt (alle diese Schritte werden vom Programm `commitUpdateStation` selbständig ausgeführt); die geänderte Konfiguration wird aktualisiert; der Job prüft noch, ob alles fehlerfrei abgelaufen war und sendet die Antwort unter dem Dateinamen `cfg.rsp.jar` zurück. Das Ergebnis dieser Änderungen wird auf der LCI in der Datei `activity.log` protokolliert.

Hinweis: Im Falle eines Misserfolgs wird auf der RCI die alte Konfiguration wieder eingespielt.

5.4 Wie verteile ich rvs® Tiny Installationen?

Dieses Kapitel beinhaltet eine Übersicht der notwendigen Schritte bei der Distribution von vielen rvs® Tiny - Installationen. Es besteht aus zwei Instruktionsteilen: zuerst werden die Schritte beschrieben, die ein rvs® Tiny-Administrator (ein LCI-Administrator, der andere rvs® Tiny-Stationen administriert, siehe Kapitel 5.1 für die Erklärung von LCI) unternehmen muss, um viele rvs® Tiny-Installationen zu verteilen; anschließend werden die Schritte beschrieben, die die Benutzer der verteilten Installationen auszuführen haben.

Dieses Kapitel kann auch benutzt werden, um die Benutzerinstruktionen an die rvs® Tiny-Stationen weiterzuleiten; man soll die ensprechende Teile des Kapitels an die eigene Systemumgebung anpassen und den Administrator-Teil entfernen.

Voraussetzung: Der Benutzer braucht Administratorrechte für die Maschine, auf welcher rvs® Tiny installiert werden soll.

Folgende Aufgaben muss der Benutzer erfüllen und die Ergebnisse an den Administrator senden:

– Öffnen Sie die Eingabeaufforderung (Kommandozeile) mit dem folgenden Befehl: `Start -> Ausführen -> cmd`.

– Geben Sie den Befehl `hostname` ein.

– Senden sie das Ergebnis (Name Ihres Computers) an den Administrator.

Nachdem der Administrator die gewünschte Angabe (Name des Computers, auf dem rvs® Tiny installiert werden soll, erhalten hat, soll er folgende Schritte durchführen:

– Einen Lizenzschlüssel von gedas für die rvs® Tiny-Remote-Station anfordern (siehe bitte Kapitel 2.2).

– eine Odette-ID für die rvs® Tiny-Remote-Station definieren oder anfordern (siehe bitte Kapitel 3.2.3).

– Odette-Passwörter mit der rvs® - Zentrale für die rvs® Tiny-Remote-Station vereinbaren (siehe bitte Kapitel 3.2.3).

– Die rvs® Tiny-Remote-Station in der eigenen Stationstabelle (von LCI) anlegen (Siehe bitte Kapitel 3.2.1).

– Die rvs® Tiny-Remote-Station in der Stationstabelle der rvs® - Zentrale anlegen (Siehe bitte Kapitel 3.2.1).

– Legen Sie eine Stationskonfigurationsdatei `rvsTinyStationlist.xml` auch für die rvs® Tiny-Remote-Station an. Dazu können Sie auch die eigene `rvsTinyStationlist.xml` benutzen, die Sie dann ensprechend mit den Daten für die rvs® Tiny-Remote-Station füllen müssen (es müssen mindestens eine lokale Station, die rvs® - Zentrale und Ihre Station, LCI, in der Stationstabelle vorhanden sein).

– Speichern Sie die folgenden Programme und Dateien auf eine CD und schicken Sie diese CD an den Benutzer der rvs® Tiny-Remote-Station.
  – JRE (Java Runtime Enviroment) 1.3 (siehe Kapitel 2.1)
  – rvs® Tiny - setup-Datei
  – den Lizenzschlüssel `licence.properties`
  – die Stationskonfigurationsdatei `rvsTinyStationlist.xml`

Nachdem der Benutzer die CD mit den Daten erhalten hat, sind folgende Aufgaben zu erfüllen, um eine funktionsfähige rvs® Tiny-Installation zu erhalten:

– Installieren Sie JRE 1.3 (Name der auszuführenden Datei auf CD `j2re-1_3_1_07-windows-i586-i.exe`).

– Installieren Sie rvs® Tiny (Name der auszuführenden Datei auf CD `rvsTiny_X-X_setup.exe`). Bei der Installation wählen Sie JRE 1.3 als Java Virtual Maschine), ansonsten beantworten Sie alle Fragen mit **Weiter**, **OK** oder **Installieren**. Notieren Sie sich den rvs® Tiny - Pfad (den Pfad wohin rvs® Tiny installiert wurde).

– Kopieren Sie den Lizenzschlüssel `licence.properties` und die Stationstabelle (Datei `rvsTinyStationslist.xml`) ins `conf`-Verzeichnis von rvs® Tiny.

– Starten Sie den rvs® Tiny Server mit dem Befehl `startServer` in der Eingabeaufforderung oder mit der rvs® Tiny - Programmgruppe (`Start -> Programme -> rvsTiny -> rvsTiny Start`). Siehe Kapitel 4.1.

© gedas deutschland GmbH / Fachbereich rvs® Systems

– Starten Sie die rvs® Tiny GUI mit dem Befehl `startGUI` in der Eingabeaufforderung oder mit der rvs® Tiny - Programmgruppe (`Start -> Programme -> rvsTiny -> rvsTiny GUI`). Siehe Kapitel 4.2.
– Notieren Sie sich die Daten der rvs® - Zentrale und der LCI-Station.
– Senden Sie die Datei `rvsTinyConfig.xml` zur LCI-Station (Format Unstrukturiert, siehe Kapitel 4.5) als Beweis für die erfolgreiche Installation und Konfiguration. Dies dient gleichzeitig als Verbindungstest.

Wenn dieser Test nicht erfolgreich sein sollte, probieren Sie nicht noch einmal, sondern kontaktieren Sie den Administrator (LCI-Station).

Der Administrator wird versuchen, den Fehler zu beheben oder im Erfolgsfall den Empfang der Datei zur Kenntnis zu nehmen und eventuell mit der Zentralen Administration fortsetzen, wenn nötig.

Symbole

$RVSTINY_HOME 7

A

activateStation 28
anpassen 15
ARCDIR 16
archiveJobs 33
Auszeichnungen 7

B

Batch-Datei 26

C

ConfigFile 15
CONTACT 18, 19, 23
createSendJob 29

D

DB 16

E

EERP 20, 24
EERP_OUT 20, 23, 24, 33
ENDED 16, 30
ENTERPRISE 19
EX_BUF_CRE 23
EX_BUF_SIZ 23

F

FAILED 16, 30

G

GateWAY_STATION_NK 24
getJob 32
getJobList 31

H

handleEERP 20, 23, 24, 33
HOLD 20, 23, 24
HostAllowFile 16
HostDenyFil 16

I

IMMEDIATE 20, 23, 24
INBOX 16
IP_ADDR 19, 23, 24

J

jobFilter 24
JobStart 24
JobstartConfigFile 16

K

Konfigurationsdateien 24

L

LINE_TYPE 22
LOCATION 19
LogConfigFil 15
LOGDIR 16

M

monlog.log 16

N

NAME 19

O

ODETTE 21
oDETTE 22, 24
ODETTE_ID 18, 19, 22, 23
Odette-Port 18

P

Plattformen 6
Port 19, 23, 24

R

RCV 16
REC_PW 19, 23
revision.log 33
RMIServiceHost 17
RMISeviceName 17
RootDir 15
Routing-Station 23
rvs® - Zentrale 22
rvsTiny.properties 15
rvsTinyConfig.xml 16
rvsTinyStationlist.xml 17

© gedas deutschland GmbH / Fachbereich rvs® Systems

S

sendAttempts 25
showMonitorLog 27
SID 18, 21, 22, 23
SN 16
SND_PW 19, 22
startServer 26
STATION_LOC 21
STATION_NK 22
STATION_RT 23
StationsConfigFile 16
stopServer 26
STREET 19
STREETNUMBER 19

T

TCPIP_BASIC 18, 19, 23, 24
TCPIP_REC 20, 21
TEMP 16
tiny.log 16

V

VDA 20
VDSN 25
vdsn 25
vorzunehmen 15

W

Was ist rvs® 5
Was rvs® nicht ist 5